

Final Report

Grant NAG5-8009

GEO MAGNETIC CUTOFF RIGIDITY COMPUTER PROGRAM
Theory, Software Description and Example

Table of Contents

Synopsis	1
Section I Science Involved	
1.1 Historical Background	2
1.2 The Equations Involved	2
1.3 Characteristics of Cosmic-Ray trajectories in the Earth's magnetic Field	5
1.4 Accuracy of the Calculations	10
1.5 Summary	10
Section II Program Descriptions	
Part 1 Description of Program TJI95	13
Part 2 Description of Program TJI95T	40
Part 3 Description of Program TJALLAMGI	67
Section III Examples	
Example of program TJI95T Execution	96
Example of TAPE1 data input file	96
Example of "on line" output of program TJI95T Execution	96
Example of TAPE7 data output file	97
Example of TAPE8 data output file	102
Section IV Rigidity to Energy Conversion	
Rigidity to Energy Conversion Text	110
Rigidity to Energy Conversion Table for $^{16}\text{O}_8$	111
Rigidity to Energy Conversion Table for $^1\text{H}_1$	112
Description of Subroutine Program AZRGEG	113
Listing of Demonstration Program ERG_RIG	114
Listing of Demonstration Program RIG_ERG	117
Program Listing	
Listing of Program TJI95	120
Listing of Program TJI95T	144
Listing of Program TJALLAMG	169

Final Report

Grant NAG5-8009

Prepared by D. F. Smart and M. A. Shea
Center for Space Plasmas and Aeronomics Research
The University of Alabama in Huntsville
Huntsville, Alabama 35889

Synopsis. The access of charged particles to the earth from space through the geomagnetic field has been of interest since the discovery of the cosmic radiation. The early cosmic ray measurements found that cosmic ray intensity was ordered by the magnetic latitude and the concept of cutoff rigidity was developed. The pioneering work of Störmer (1930, 1955) resulted in the theory of particle motion in the geomagnetic field, but the fundamental mathematical equations developed have "no solution in closed form". This difficulty has forced researchers to use the "brute force" technique of numerical integration of individual trajectories to ascertain the behavior of trajectory families or groups. This requires that many of trajectories must be traced in order to determine what energy (or rigidity) a charged particle must have to penetrate the magnetic field and arrive at a specified position. It turned out the cutoff rigidity was not a simple quantity but had many unanticipated complexities that required many hundreds if not thousands of individual trajectory calculations to resolve. The accurate calculation of particle trajectories in the earth's magnetic field is a fundamental problem that limited the efficient utilization of cosmic ray measurements during the early years of cosmic ray research.

As the power of computers has improved over the decades, the numerical integration procedure has grown more tractable, and magnetic field models of increasing accuracy and complexity have been utilized. These improvements have made the general application of the numerical integration procedure more practicable and while the cutoff rigidity problem is still formidable, thousands of trajectories can be computed without the expenditure of excessive resources.

This report is documentation of a general FORTRAN computer program to trace the trajectory of a charged particle of a specified rigidity (momentum per unit charge) from a specified position and direction through a model of the geomagnetic field. This software has been incorporated into a general control program that makes the computation of a number of trajectories to scan through a rigidity interval to determine the cutoff rigidity of a specified location. The input control file may contain as many locations as deemed necessary for a specific study.

This report is organized in sections. Section I gives a scientific background. Section II gives program documentation for three versions of the program. Section III provides examples of the data input and examples of the program output for the station selected. Section IV is an appendix describing rigidity to energy conversion with tables and demonstration programs. The final section is a listing of the trajectory program FORTRAN source codes (with added line numbers).

This document is also included on a IOMEGA 100 MB ZIP disk, and the FORTRAN source code is also provided on a 1.44 MB 'floppy' disk.

1.1 Historical Background

The integration of the equation of motion of a charged particle in a magnetic field is a problem that has no solution in a closed form. The first numerical efforts at integration of the equations of particle motion began with Störmer (1930) who utilized a dipole representation of the earth's magnetic field. The work of Störmer is summarized in his book 'The Polar Aurora' (Störmer, 1950). Lemaître and Vallarta (1936 a,b) used a "Bush differential analyzer" (what would now be called an analog computer) to obtain solutions for entire families of trajectories. Jory (1956), Lust (1957), and Kasper (1959) were among the first researchers to utilize the digital computer as a tool for trajectory calculations in a dipole magnetic field. More advanced magnetic field models were utilized by McCracken and his co-workers (McCracken *et al.*, 1962, 1965, 1968). These workers were very successful in the use of high speed digital computers for the calculation of cosmic ray trajectories in high order simulations of the geomagnetic field. They calculated particle access to specific cosmic ray stations on the earth to describe the cosmic ray anisotropy and also showed that the observed cosmic ray intensity could be well ordered by geomagnetic cutoff rigidities derived from cosmic ray trajectories calculated in high order simulations of the earth's magnetic field (Shea *et al.*, 1965). They also demonstrated that the earth's internal magnetic field is evolving (quite rapidly on geologic time scales), and that the use of updated magnetic field models is necessary to explain the changes observed in cosmic ray intensity in some areas of the world (Shea and Smart, 1970, 1990; Mischke *et al.*, 1979). This is necessary because the earth's geomagnetic field evolution is not uniform, and sudden changes (called geomagnetic "jerks") have been found in the Earth's magnetic field (Langel *et al.*, 1986; Macmillan, 1996).

Advances in computer technology over the past decades have allowed researchers to more fully utilize the trajectory-tracing technique. As computers become more powerful, magnetic field models of increasing complexity, which better represent the earth's magnetic topology, have been developed and must be utilized for analyses of the higher precision measurements of cosmic radiation phenomena. As long as the measurement techniques increase in accuracy and as long as the geomagnetic field models continue to improve, the trajectory-tracing process will be used for cosmic radiation research.

1.2. The Equations Involved

1.2.1 The Charged Particle Equation of Motion

The equation of charged particle motion in a magnetic field may be written in vector form as

$$\ddot{\mathbf{r}} = (e/mc) \dot{\mathbf{r}} \times \mathbf{B}.$$

In this equation, $\ddot{\mathbf{r}}$ is the particle acceleration, $\dot{\mathbf{r}}$ the particle velocity, and \mathbf{B} the magnetic field vector. The electronic charge is denoted by e , m is the particle's relativistic mass, and c is the speed of light. This equation, when expressed in r, θ, ϕ coordinates, results in three simultaneous differential equations with six unknowns.

$$\frac{dv_r}{dt} = \frac{e(v_\theta B_\phi - v_\phi B_\theta)}{mc} + \frac{v_\theta^2}{r} + \frac{v_\phi^2}{r}$$

$$\frac{dv_\theta}{dt} = \frac{e(v_\phi B_r - v_r B_\phi)}{mc} - \frac{v_r v_\theta}{r} + \frac{v_\phi^2}{r \tan\theta}$$

$$\frac{dv_\phi}{dt} = \frac{e}{mc} (v_r B_\theta - v_\theta B_r) - \frac{v_r v_\phi}{r} - \frac{v_\theta v_\phi}{r \tan\theta}$$

In these equations the particle velocity terms are

$$\frac{dr}{dt} = v_r$$

$$\frac{d\theta}{dt} = \frac{v_\theta}{r}$$

$$\frac{d\phi}{dt} = \frac{v_\phi}{r \sin\theta}$$

This system of simultaneous linear differential equations can be integrated numerically if the components of magnetic induction B_r, B_θ, B_ϕ , are known as explicit functions of r, θ, ϕ . The method chosen by McCracken *et al.* (1962) to solve the above system of equations was fourth order Runge-Kutta integration (Ralston and Wilf, 1960). In this numerical integration process, when the magnetic field is known (see next section), a knowledge of the position and velocity coordinates on one point of the trajectory is used with the differential equations of motion to give the coordinates of subsequent points along the trajectory. Repeated application gives sufficient points to locate the trajectory in space. Adaptive step size control (see section 1.2.3.1) can make the process more efficient. This is sometimes called fifth order Runge-Kutta (see Press *et al.*, 1989).

1.2.2 Computing the Earth's Magnetic Field

Computation of a high order simulation of the earth's magnetic field is a computer intensive process and to the surprise of many, even more demanding of computer resources than integration of particle trajectories.

If the field being modeled is composed of only internal sources, then it is possible to define a magnetic potential, V , that can be expanded in spherical harmonics.

$$V(r, \theta, \phi) = a \sum_{n=1}^{\infty} (a/r)^{n+1} \sum_{m=0}^n [g_m^n \cos m\phi + h_m^n \sin m\phi] P_m^n(\cos \theta)$$

In this equation g_m^n and h_m^n are the Gauss coefficients describing the magnetic field, $P_m^n(\cos \theta)$ are the Schmidt-normalized associated Legendre polynomials, and a is the average radius of the earth. In the dipole case, the expansion results in simple algebraic equations in r, θ, ϕ that can be repeatedly evaluated to quickly find a solution for a specific trajectory initiated from a specified direction at a specific energy. However, as the complexity of the magnetic field expansion increases, the number of terms to be evaluated increases as $n!$. For a 10th order description of the earth's main magnetic field as provided by the International Geomagnetic Reference Field (IGRF, 1992; Sabaka, 1997), about 90 percent of the computer processing time is consumed in evaluating the magnetic field and only about 10 percent of the CPU time utilized in integrating the particle equation of motion. The most efficient computer techniques available for evaluating the Legendre polynomial expansion involve using the

derivative of the previous term to obtain the current term, a process that is inherently serial. The use of the recursion process is about an order of magnitude slower. (All attempts to develop a very efficient parallel-processing algorithm to evaluate magnetic fields have so far met with failure.)

1.2.3 Methods for Efficient Computation of Cosmic-Ray Trajectories.

It is difficult to calculate the trajectory of an incoming cosmic ray particle through the magnetic field and expect to intersect the exact location for which the calculation was desired. Since the path of a negatively charged particle of a specific magnetic rigidity is identical (except for the sign of the velocity vector) to that of a positively charged particle reaching the same location in space, the common method of calculating cosmic ray trajectories in the earth's magnetic field is to calculate the trajectory in the reverse direction. Thus for cosmic ray trajectory calculations the "starting point" of the reverse trajectory calculation is given by the geographic coordinates, direction and altitude of the location in question.

The extreme requirement of intensive computation to obtain a sufficient number of particle trajectories to evaluate cosmic ray access to a specific location on the earth or in the earth's magnetosphere may involve obtaining solutions to millions of individual cosmic ray trajectories. Therefore efficient computation is essential (and a fast computer desirable).

1.2.3.1 Variable Step Size Methods for Computation of Cosmic-Ray Trajectories.

One approach developed by Smart and Shea (1981a) was to compute a dynamic variable step length that was of the order of one percent of a particle gyro-distance in the magnetic field. This process allows computation of a simple cosmic ray trajectory from the "top" of the atmosphere to interplanetary space in about 100 Runge-Kutta iterations. Complex trajectories, or trajectories of low rigidity (rigidity is momentum per unit charge) take correspondingly more iterations. The gyro-radius of a charged particle in a magnetic field is given by

$$\rho = 33.33 R / B.$$

In this equation ρ is the particle gyro-radius in km, R is the particle rigidity in units of GV, and B is the magnitude of the magnetic field in units of Gauss.

The particle velocity can be specified as the ratio of the particle speed to the speed of light (v/c) and designated by the symbol β which can be derived from the relativistic factor, γ , as follows:

$$\beta = [1.0 - (1.0/\gamma^2)]^{1/2},$$

and

$$\gamma = \{[(R Z)/(m_0 c^2 A)]^2 + 1.0\}^{1/2},$$

where R is the particle rigidity¹, Z the atomic charge, A the atomic number and $m_0 c^2$ is the rest mass energy.

¹ Rigidity is momentum per unit charge and is a canonical unit that is especially useful in characterizing charged particle access in magnetic fields. All particles having the same magnetic rigidity, charge sign and initial conditions will have identical trajectories in the magnetic field, independent of elemental or isotopic composition, particle mass or atomic charge.

1.3. Characteristics of Cosmic-Ray Trajectories in the Earth's Magnetic Field

To examine the characteristic behavior of cosmic ray trajectories in the earth's magnetic field we consider trajectories of cosmic ray particles with different energies as these trajectories are calculated from a location at the top of the atmosphere outward into the magnetic environment surrounding the earth. The trajectory for a very high-energy particle propagating outward through the earth's magnetic field will reach interplanetary space with a minimum of geomagnetic bending. As the charged particle energy decreases, then it will undergo more geomagnetic bending before it can escape. At some lower energy, it will no longer have sufficient momentum to escape the magnetic field and in these cases the particle trajectory initiated in an outward direction near the top of the atmosphere, will re-enter (i.e. intersect the solid earth). The presence of a solid object in the magnetic field complicates the problem, and an analytical description of the phenomena becomes even more complicated if the solid object is not centered in the magnetic field.

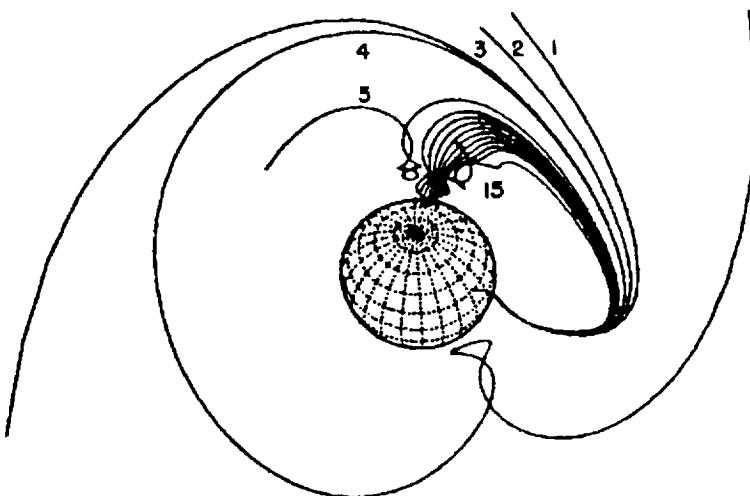


Figure 1. Illustration of charged particle trajectories of different energies (rigidities) traced out in the vertical direction from the same location. The trajectories undergo increased geomagnetic bending as the particle energy (rigidity) is decreased. Charged particle trajectories near the cutoff rigidity develop intermediate loops and become complex. In the cosmic ray penumbra, some trajectories are re-entrant, and some are allowed. See text for more details.

Some actual trajectory calculations are illustrated in Figure 1. All of the trajectories in this Figure were initiated in the vertical direction from the same location. The trajectories labeled 1, 2, and 3 show increasing geomagnetic bending before escaping into space. The trajectory labeled 4 develops intermediate loops before escaping. The lower energy trajectory labeled 5 develops complex loops near the earth before it escapes. As the charged particle energy is further reduced, there are a series of trajectories that intersect the earth (i.e. re-entrant trajectories). In a pure dipole field that does not have a physical barrier embedded in the field, these trajectories may be allowed, illustrating one of the differences between Störmer theory and trajectory calculations in the earth's magnetic field. Finally the still lower energy trajectory labeled 15 escapes after a series of complex loops near the earth. These series of allowed and forbidden bands of particle access are called the cosmic ray penumbra. They also illustrate an often-ignored fact that cosmic ray geomagnetic cutoffs are not sharp (except for special cases in the equatorial regions).

1.3.1 Cutoff Rigidities

Our procedure for determining geomagnetic cutoff rigidities is to make trajectory calculations at discrete intervals through the rigidity spectrum with the assumption that the results of a specific trajectory at a specific rigidity are characteristic of adjacent trajectories at very slightly different rigidities or direction. These calculations begin at high rigidities (at a value above the highest possible cutoff) and progress down through the rigidity spectrum until the lowest possible allowed trajectory has been found. An examination of the characteristics of particle trajectories from high rigidities to low rigidities will show definitive fiducial marks. These are the first discontinuity in asymptotic direction, the first forbidden trajectory, and perhaps a range of allowed and forbidden trajectories called the cosmic ray penumbra, and the lowest allowed trajectory. In the cosmic ray penumbra, the highest rigidity forbidden band is called the "first forbidden band" (see Smart *et al.*, 2000, for more discussion). We currently use three parameters to describe a geomagnetic cutoff rigidity. These are:

- R_u The upper cutoff which is the rigidity of the last allowed before the first forbidden trajectory,
- R_l The lower cutoff which is the rigidity of the last allowed trajectory in a decreasing rigidity scan, and
- R_c The effective cutoff which is an average between R_u and R_l that accounts for the transparency of the penumbra.

A more detailed explanation of the characteristics of geomagnetic cutoffs derived from trajectory calculations is given by Cooke *et al.* (1991). Figure 2 illustrates cosmic ray penumbra structure and geomagnetic cutoffs determined by trajectory calculations for three North American neutron monitor stations.

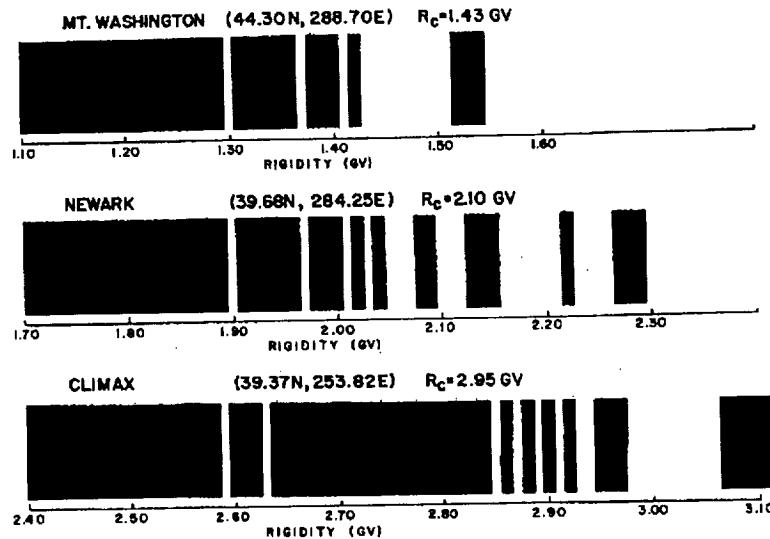


Figure 2. Illustration of trajectory-derived cosmic ray cutoff and the cosmic ray penumbra structure in the vertical direction. The calculations have been done for three North American neutron monitor stations. White indicates allowed rigidities, black indicates forbidden rigidities.

Since there are chaotic structures in the penumbral region with very small features there is no certainty that all features are identified in a rigidity scan. It is possible that we might not identify very small penumbral bands near the cutoff. When scanning the asymptotic directions that represent the interplanetary terminus of these trajectory calculations as a function of rigidity, there is a systematic increase in asymptotic longitude as the rigidity is decreased, until very near the cutoff there is a

discontinuity in asymptotic direction. We have found that whenever there is a discontinuity in asymptotic direction and we investigate the rigidity region in minute detail, there is a forbidden (re-entrant) trajectory associated with the discontinuity. Therefore, the first discontinuity in asymptotic direction is always the start of the penumbra. Continuing downward through the penumbra and calculating trajectories for particles having successively lower rigidities results in a last allowed trajectory that identifies the lower rigidity end of the cosmic ray penumbra.

1.3.2 Asymptotic Directions of Approach

If we follow a charged particle trajectory away from the earth, the amount of geomagnetic bending per unit path length decreases. In a magnetic field extending to infinity, it can be said that the particle direction asymptotically approaches its final direction. If we introduce a boundary such as the magnetopause, we often use the same terms to describe the direction of the particle velocity vector at the penetration location. (Ruth Gall in her work was most specific that these were directions of approach.) McCracken and co-workers (McCracken *et al.*, 1968; Shea *et al.*, 1965), performed calculations in internal magnetic fields and utilized the particle velocity vector (expressed in geocentric coordinates at radial distance of 25 earth radii) to specify the asymptotic direction of approach. The set of asymptotic directions accessible to a specific location on the earth defines the asymptotic cone of acceptance. The asymptotic longitude can exceed 360 degrees. Large asymptotic longitudes are indicative of how many times the trajectory has circumnavigated the earth during its transit.

In early work on trajectory calculations Kasper (1959) found the "focusing effect" of the magnetic field where trajectories initiated outward from the earth with different azimuth and zenith angles of incidence (at high latitudes, within a factor of two above the cutoff rigidity) reached a similar final asymptotic direction at distances far from the earth. This "focusing effect" which is valid when the scale size of the gradient in the earth's magnetic field is less than the particle gyro-radii, also leads to the concept that asymptotic directions computed for vertically arriving particles are a good approximation of the entire asymptotic cone of acceptance.

For polar or even mid-latitude muon detectors that only respond to high-energy particles, these asymptotic cones of acceptance are restricted to specific regions of the celestial sphere. Thus if multiple stations simultaneously observe an anisotropic solar cosmic ray flux, it is possible to deconvolve the flux direction in space and the anisotropy (see Cramp *et al.*, 1995). If these stations are located at different geomagnetic cutoffs, it is possible to deduce the solar particle spectra. Similarly, if a number of cosmic ray stations, each having asymptotic cones of acceptance viewing a different portion of the celestial sphere, rotate through a slowly evolving cosmic ray anisotropy, then it is possible to deconvolve the spatial anisotropy. (See Nagashima and Fujimoto, 1994, for an example of this application.) The asymptotic directions of approach in the rigidity range from 20 GV to 5 GV computed for cosmic ray muon detectors for the maximum of the 29 September 1989 high-energy solar cosmic ray events are illustrated in Figure 3.

In a rigidity scan of the trajectories allowed at a specific location (cosmic ray detector) the geomagnetic bending of the particle trajectory increases as the particle rigidity decreases as illustrated in Figure 1. The amount of geomagnetic bending becomes very large as the particle rigidity approaches the geomagnetic cutoff rigidity, perhaps involving several circum-navigations of the earth. The result is an extremely broad asymptotic cone of acceptance for mid- or low-latitude stations with a large range of asymptotic longitudes involved. Figure 5 illustrates asymptotic cones of acceptance for selected neutron monitor stations projected on a spherical mapping of the earth. Note the longitudinal extent of the asymptotic cones for the Calgary, Deep River, and Goose Bay, Canada and the Hobart, Australia cosmic ray stations.

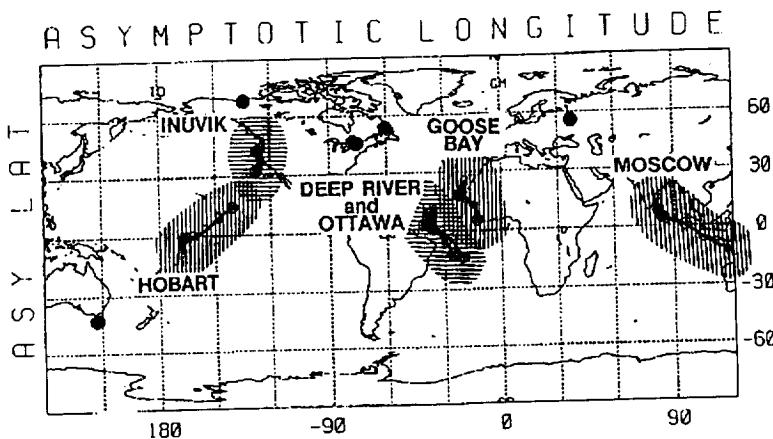


Figure 3. World map projection of the asymptotic directions of approach computed for cosmic ray muon detectors for the 29 September 1989 high-energy solar cosmic ray events.

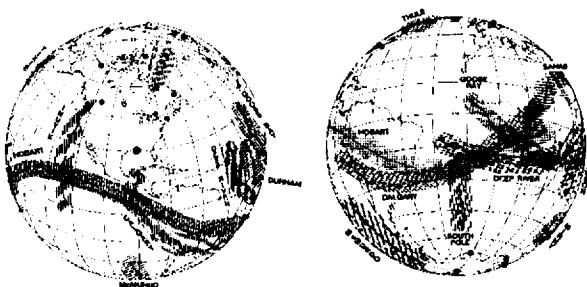


Figure 4. Asymptotic directions of approach computed for selected cosmic ray neutron monitors mapped on a spherical projection of the earth. These projections are oriented on the probable interplanetary magnetic field direction for two specific solar cosmic ray events.
Left: 29 September 1989. Right: 19 October 1989.

In the trajectory calculations, we compute a trajectory at a specific rigidity and direction and then assume that this result is representative of a finite domain of rigidity or angular space. There is the possibility that sampling the rigidity spectrum at uniform intervals such as 0.01 GV might not identify the first transition from the continuously allowed rigidities to the cosmic ray penumbral regions of alternating allowed and forbidden rigidity bands. There is also the question of how valid is the approximation that a sample in one direction is truly representative of the entire asymptotic cone of acceptance for a wide variety of directions. We have no definitive answer as yet to these questions.

The problem of determining which trajectories are allowed and which are forbidden is not as simple as it might initially seem. In the internal magnetic field representations and especially in the more complex magnetospheric fields, there is a set of low rigidity trajectories that has very long path lengths, consisting of many complex loops. Often for the sake of economy of computer resources, trajectory calculations are terminated after a large number of steps. This results in groups of indeterminate trajectories whose fate is not resolved. In Störmer theory there is a special set of trajectories which will have an arbitrary number of loops before reaching a final solution. In a simple dipole field, these low

rigidity trajectories having many loops were generally forbidden. Shea *et al.* (1965) adopted the convention of declaring these indeterminate solutions as forbidden. This convention is questionable, especially since these trajectory paths are the result of a stable magnetic field and the magnetosphere is a domain of dynamic plasma processes. Lin *et al.* (1995) found that their result of charged particle access to a cosmic ray detector in a balloon flown at high latitudes was consistent with defining these low rigidity indeterminate trajectories as representing allowed charged particle access through the earth's magnetosphere. Boberg *et al.* (1995) considered any trajectory that originated at low altitudes and reached the altitude of a geosynchronous satellite to be allowed. Tylka *et al.* (1995) and Smart *et al.* (1999a,b,c) adopted the Boberg *et al.* (1995) definition in their recent work for calculating geomagnetic cutoff rigidities.

However, there are definite limits to the use of the vertically incident cosmic ray trajectories to provide an "exact" cutoff rigidity. The 'pencil-thin' particle beam being simulated may encounter a penumbral structure that is not truly representative of the cosmic ray access over a wider solid angle of acceptance. This leads to a 'lumpy' structure that may not properly order the counting rate acquired by a neutron monitor during a latitude survey. However, the time requirements of computing a complete world grid of cutoff rigidities for a variety of directions has been so formidable that the vertical cutoff approximation is the most widely used set of cutoff rigidities. Figure 5 shows the result of a trajectory derived vertical cutoff rigidity at one-degree intervals along the 285 degree East meridian from the cosmic ray knee to the cosmic ray equator. Note the irregular character of the calculated cutoff values.

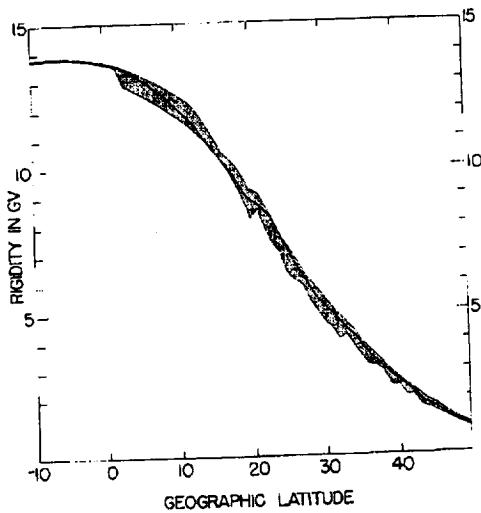


Figure 5. A set of trajectory derived vertical cutoff rigidity values, calculated at one degree intervals, along the 285-degree east meridian from the cosmic ray knee to the cosmic ray equator. Note the irregular character of the cutoff values. The upper computed cutoff, R_u , is indicated by the upper boundary of the shaded area; the lower computed cutoff, R_l , (the last allowed trajectory) is indicated by the lower boundary of the shaded area. The solid line is the 'effective cutoff', R_c , attempting to account for the transparency of the penumbra in the method as defined by Shea *et al.* (1965).

1.4. Accuracy of the Calculations

The accuracy of the magnetic field models employed is the limiting factor in charged particle trajectory calculations assuming that the numerical techniques yield an exact solution and the computers involved have sufficient numerical accuracy. The high order simulations of the earth's magnetic field are better representations than the simple models.

For precise trajectories involving exact locations on the earth, then the initial directions must be specified in geodetic coordinates. See for example, Smart and Shea (1981b) for calculation of the termination of muon trajectories from the Batavia, Illinois, USA high-energy particle accelerator. For this mid-latitude location, the geodetic horizon is at an elevation angle of about $\frac{1}{2}$ degree when transformed into geocentric coordinates. Shea and Smart (1983), Shea *et al.* (1987), and Smart and Shea (1997a) use geodetic coordinates when calculating cutoff rigidities for locations on the surface of the earth or in the earth's atmosphere, but use geocentric coordinates when calculating particle access or geomagnetic cutoff for spacecraft (Smart and Shea, 1997b). We have found a few noxious cases where, in complex particle trajectories near the cutoff rigidity, there were sudden, very small loops in the trajectory and the step size adjustment algorithm did not respond with sufficient agility to faithfully trace the trajectory. However, these cases are relatively rare. (The classic method to check the accuracy of a numerical integration procedure is to halve the step length, repeat the calculation, and verify that the same solution is obtained.)

Some experimenters such as Dryer and Meyer (1975) have used the prediction of the geomagnetic cutoff derived from trajectory calculations in the design of experiments that respond to cosmic ray heavy nuclei in a specific rigidity range. These attempts have been very successful indicating that there is a general reliability in high-energy charged particle trajectory calculations in high degree simulations of the earth's magnetic field.

1.5. Summary

The calculation of particle trajectories in the earth's magnetic field was a fundamental problem that limited the efficient utilization of cosmic ray measurements during the early years of cosmic ray research. As the power of computers has improved over the decades, the numerical integration procedure has grown more tractable, and magnetic field models of increasing accuracy and complexity can be utilized. The trajectory calculation process is sufficiently mature that it is possible to do sufficient trajectory calculation to determine and cutoff rigidities. It is now possible for experiments to be designed on the basis of trajectory calculations.

References

- Boberg, P.R., Tylka, A.J., Adams, J.H., Flückiger, E.O., and Kobel, E.: 1995, 'Geomagnetic Transmission of Solar Energetic Protons During the Geomagnetic Disturbances of October 1989', *Geophys. Res. Lett.* **22**, 1133-1136.
Cooke, D.J., Humble, J.E., Shea, M.A., Smart, D.F., Lund, N., Rasmussen, I.L., Byrnak, B., Goret, P and Petrou, N.: 1991, 'On Cosmic-Ray Cutoff Terminology', *Il Nuovo Cimento* **14C**, 213-234.
Cramp, J.L., Duldig, M.L., and Humble, J.E.: 1995, 'Neutron Monitor Response to Highly Anisotropic Ground Level Enhancements', *Proc. Int. Cosmic Ray Conf.* **24th** **4**, 248-251.
Dryer, R., and Meyer, P.: 1975, 'Isotopic Composition of Cosmic-Ray Nitrogen at 1.5 GeV/Amu', *Phys. Rev. Lett.* **35**, 601-604.
IGRF: 1992, 'IGRF, 1991 Revision', *EOS, Trans. American Geophys. Union*, **73** No.16, 182.
Jory, F.S.: 1956, 'Selected Cosmic-Ray Orbits in the Earth's Magnetic Field', *Phys. Rev.* **103**, 1068-1075.
Kasper, J.K.: 1959, 'The Earth's Simple Shadow Effect on Cosmic Radiation', *Nuovo Cimento* **XI**, (Supplemento) 1-26.

- Langel, R.A., Kerridge, D.R., Barraclough, D.R., and Main, R.C.: 1986, 'Geomagnetic Temporal Change: 1903-1982', *J. Geomag. Geoelectr.* **38**, 573-597.
- Lemaire, G., and Vallarta, M.S.: 1936a, 'On the Geomagnetic Analysis of Cosmic Radiation', *Phys. Rev.* **49**, 719-726.
- Lemaire, G., and Vallarta, M.S.: 1936b, 'On the Allowed Cone of Cosmic Radiation', *Phys. Rev.* **50**, 493-504.
- Lin, Z., Bieber, J., and Evenson, P.: 1995, 'Electron Trajectories in a Model Magnetosphere: Simulation and Observations under Active Conditions', *J. Geophys. Res.*, **100**, 23543-23549.
- Lust, R.: 1957, 'Impact Zones for Solar Cosmic Ray Particles', *Phys. Rev.* **105**, 1827-1839.
- Macmillan, S.: 1996, 'A Geomagnetic Jerk for the Early 1990's', *Earth Planet. Sci. Lett.* **13**, 189-192.
- McCracken, K.G., Rao, U.R., and Shea, M.A.: 1962, 'The Trajectories of Cosmic Rays in a High Degree Simulation of the Geomagnetic Field', Massachusetts Institute of Technology, Laboratory for Nuclear Science, Technical Report 77.
- McCracken, K.G., Rao, U.R., Fowler, B.C., Shea, M.A., and Smart, D.F.: 1965, 'Cosmic Ray Tables (*Asymptotic directions, variational coefficients and cutoff rigidities*)', *IQSY Instruction Manual No. 10*, IQSY Committee, London.
- McCracken, K.G., Rao, U.R., Fowler, B.C., Shea, M.A., and Smart, D.F.: 1968, 'Cosmic Ray Tables (Asymptotic Directions, etc.)', *Annals of the IQSY, I, Chapter 14*, 198-214. MIT Press, Cambridge.
- Mischke, C.F.W., Raubenheimer, B.C., Stoker, P.H., van der Walt, A.J., Shea, M.A., and Smart, D.F.: 1979, 'Experimental Observations of Secular Changes in the Vertical Cutoff Rigidity', *Proc. Int. Conf. Cosmic Ray 16th* **4**, 279-284.
- Nagashima, G., and Fujimoto, K.: 1994, 'Interplanetary Magnetic Field Collimated Cosmic Ray Flow Across Magnetic Shock From Inside of Forbush Decrease, Observed as Local-Time-Dependant Precursory Decrease on the Ground', *J. Geophys. Res.* **99**, 21419-21427.
- Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: 1989, 'Numerical Recipes', Cambridge University Press, London.
- Ralston, A., and Wilf, S.H., Eds.: 1960, 'Mathematical Methods for Digital Computers', John Wiley and Sons, New York.
- Sabaka, T.J., Langel, R.A., Baldwin, R.T., and Conrad, J.A.: 1997, 'The Geomagnetic Field, 1900-1995, Including the Large Scale Fields from Magnetospheric Sources and NASA Candidate Models for the 1995 Revision of the IGRF', *J. Geomag. Geoelectr.* **49**, 157-206.
- Shea, M.A., and Smart, D.F.: 1970, 'Secular Variation in Cosmic Ray Cutoff Rigidities', *J. Geophys. Res.* **75**, 3921-3922.
- Shea, M.A., and Smart, D.F.: 1983, 'A World Grid of Calculated Vertical Cutoff Rigidities for 1980.', *Proc. Int. Cosmic Ray Conf. 18th* **3**, 415-418.
- Shea, M.A., and Smart, D.F.: 1990, 'The Influence of the Changing Geomagnetic Field on Cosmic Ray Cutoff Rigidities', *J. Geomag. Geoelectr.* **42**, 1107-1121.
- Shea, M.A., Smart, D.F., and Gentile, L.C.: 1987, 'Vertical Cutoff Rigidities Calculated From the Estimated 1985 Geomagnetic Field Coefficients', *Proc. Int. Cosmic Ray Conf. 20th* **4**, 205-207.
- Shea, M.A., Smart, D.F., and McCracken, K.G.: 1965, 'A Study of Vertical Cutoff Rigidities Using Sixth Degree Simulations of the Geomagnetic Field', *J. Geophys. Res.* **70**, 4117-4130.
- Smart, D.F., and Shea, M.A.: 1981a, 'Optimum Step Length Control for Cosmic Ray Trajectory Calculations', *Proc. Int. Cosmic Ray Conf. 17th* **4**, 255-258.
- Smart, D.F., and Shea, M.A.: 1981b, 'Muon Trajectories from the Batavia Accelerator N-E Beam Dump', *Proc. Int. Cosmic Ray Conf. 17th* **5**, 6-9.
- Smart, D.F., and Shea, M.A.: 1997a, 'World Grid of Cosmic Ray Vertical Cutoff Rigidities for Epoch 1990.0', *Proc. Int. Cosmic Ray Conf. 25th* **2**, 401-404.
- Smart, D.F., and Shea, M.A.: 1997b, 'Calculated Cosmic Ray Cutoff Rigidities at 450 km for Epoch 1990.0', *Proc. Int. Cosmic Ray Conf. 25th* **2**, 397-400.
- Smart, D.F., Shea, M.A., and Flückiger, E.O.: 1999a, 'Calculated Vertical Cutoff Rigidities for the International Space Station During Magnetically Quiet Times', *Proc. Int. Cosmic Ray Conf. 26th* **7**, 394-397.
- Smart, D.F., Shea, M.A., and Flückiger, E.O.: 2000, 'Magnetospheric Models and Trajectories Calculations', *Space Science Rev.*, **93**, 281-308.
- Smart, D.F., Shea, M.A., Flückiger, E.O., Tylka, A.J., and Boberg, P.R.: 1999b, 'Calculated Vertical Cutoff Rigidities for the International Space Station During Magnetically Active Times', *Proc. Int. Cosmic Ray Conf. 26th* **7**, 398-401.

- Smart, D.F., Shea, M.A., Flückiger, E.O., Tylka, A.J., and Boberg, P.R.: 1999c, 'Changes in Calculated Vertical Cutoff Rigidities at the Altitude of the International Space Station as a Function of Magnetic Activity', *Proc. Int. Cosmic Ray Conf.* 26th 7, 337-340.
- Störmer, C.: 1930, 'Periodische Elektronenbahnen im Feld lines Elementramagneton und ihre Anwendung auf Bruches Modellverauche und auf Eschenhagens Elementarwellen des Erdmagnetismus'. *Zeits f. Astrophys.* 1 237-274.
- Störmer, C.: 1950, '*The Polar Aurora*' Oxford University Press, London.
- Tylka., A.J., Boberg, P.R., Adams, J.H., and Beahm, L.P.: 1995, 'The Mean Ionic Charge State of Solar Energetic Fe Ions Above 200 MeV per Nucleon', *Astophys. J. Lett.*, 444, L109-L113.

DESCRIPTION OF PROGRAM TJI95

This software package is self-contained and capable of being compiled and executed on a variety of platforms ranging from a personal computer to large scale "super computers". The software is written in FORTRAN 77. The software is designed to efficiently compute the trajectory of an energetic charged particle of a specified momentum per unit charge (rigidity) through a model magnetic field. For cosmic ray access to the earth, the geocenter becomes the origin of the coordinate system. All calculations are done in the r, θ, ϕ coordinate system (a right-handed, orthogonal coordinate system). The magnetic field subroutine included in this software is designed for efficient evaluation of the IGRF95 model of the earth's magnetic field.

In its usual mode of computing the path of cosmic ray trajectories in a model of the earth's magnetic field, we utilize this program to determine the path of a cosmic ray (a positively charged particle) from interplanetary space arriving at the earth at a specified position and direction. To accomplish this, a negatively charged particle is 'launched' from the 'top' of the atmosphere at a specified position (latitude and longitude) in a specific direction (zenith and azimuth), and its path traced though the model magnetic field until it either (1) reaches a specified radial distance, (2) reenters the atmosphere, or (3) fails to reach either condition by a specified number of iterative steps. If the negative test particle path penetrates the specified outer boundary (reaches interplanetary space) the direction of the particle velocity vector at the boundary crossing is specified as asymptotic latitude and longitude (in corresponding geocentric coordinates). If the charged particle re-enters the atmosphere, then the re-entrant coordinates (geocentric latitude and longitude) are given. In this version of the software, an oxygen nuclei (^{16}O) is used as the test particle. Since rigidity is a canonical coordinate, the path of any charged particle having the specified rigidity will be the same.

The software can be adapted to trace the path of any particle of a specified rigidity from a specified position and direction through any model magnetic field as long as the magnetic field is expressed as vectors in the r, θ, ϕ coordinate system.

The 'input' to this program is a data line that specifies the initial position (latitude, longitude and altitude above the earth's surface), direction (azimuth and zenith), and rigidity (momentum per unit charge) along with control parameters to do 'N' trajectory calculations at specified rigidity increments beginning at an initial specified rigidity.

The 'output' of the software is in summary files with one line for each trajectory calculation.

The traditional summary output is called TAPE7 and is in an 80-column "card image" format. This contains the initial conditions (the geodetic latitude, longitude, rigidity, zenith, and azimuth), the final results (asymptotic latitude and longitude), fate of the trajectory, the number of steps in the trajectory calculation, plus a magnetic field identifier.

There is also a second output summary, traditionally called TAPE8, which is in a line printer 132 column format. This contains more detail; the initial conditions (the geodetic and geocentric latitude, the longitude, rigidity, zenith, and azimuth), the final results (asymptotic latitude and longitude, path length (in earth radii), trajectory transit time, time at altitudes under 100 km, number of maximum and minimum in radial distance along the trajectory, the trajectory fate, the number of steps in the calculation, plus a magnetic field identifier.

This structured FORTRAN 77 software assembly consists of a main program and four associated subroutines. The software has extensive internal comments to aid the user in understanding the program. The program and subroutines are:

Program	TJI95	Main program; primary purpose is control
Subroutine	GDGC	Conversion from geodetic to geocentric coordinates
Subroutine	SINGLTJ	Calculates a particle trajectory
Subroutine	FGRAD	Evaluates the $\mathbf{V} \times \mathbf{B}$ force vectors on the particle
Subroutine	MAGNEW95	Evaluates the vector magnetic field at position r, θ, ϕ

Program Organization:

Each subroutine has a separate unique function. Critical and often used variables are defined in labeled common blocks. The important "working" variables are in the common block WRKVLU. The trigonometric sines and cosines are in common block WRKTSC. Definitions associated with the shape of the ellipsoid representing the surface of the earth are in common block GEOID. We have found that some "super computers" do not allow mixing of real and integer variables in the same common block; therefore there are two additional common blocks associated with subroutine SINGLTJ. These are common block SNGLR (real variables) and common block SNGLI (integer variables)

Accuracy and Precision

It is recommended that the REAL*8 precision always be used. The primary limitation affecting the results is the accuracy of the magnetic field expansion. For reasonably simple trajectories the results should be repeatable, independent of the computer platform used. For long complex trajectories, default compiler options (round off or truncate, and the precision of intrinsic function) begin to affect the result. The calculation procedure includes automatic error checking. The particle acceleration terms are monitored in subroutine SINGLTJ. When significant increases in the force on the particle are noted the step size is reduced and the calculations are continued at smaller step intervals. The quantity BETA ($\beta = v/c$) should be invariant throughout the calculation and is monitored. Changes of BETA exceeding 1 part in 10^5 results in an automatic restart and the trajectory is recalculated at smaller step size increments.

On some "main frames" the intrinsic functions are automatically derived in REAL*8 precision; on some other systems the intrinsic functions are evaluated in a REAL*4 mode unless the double precision argument is specified. In this version intended for the Desktop Computer, all intrinsic functions are specified in the double precision mode; however, we have left a single precision statement "commented out" immediately before each double precision statement. For simple trajectories the user probably cannot note any difference; however, for long complex trajectories, the differences between the use of single precision intrinsic functions and double precision intrinsic functions will become apparent. If a specific rigidity and direction is used for comparison and the position of each trajectory step is monitored, the effect of the small differences between REAL*4 and REAL*8 accumulate and eventually the trajectory path will differ if it is a long complex trajectory.

In the interest of computational speed, the magnetic field calculation routine drops the evaluation of the high order terms when they make an "insignificant" contribution to the total magnetic field. Again for long complex trajectories, these small differences accumulate and the trajectory paths may diverge when different criteria are used for dropping magnetic field expansion.

User Defined Parameters:

Two variables are intended to be user defined. These are FSTEP and LIMIT. Default values have been set in the program, LIMIT = 600,000, and FSTEP = 4×10^8 .

LIMIT is the number of Runge-Kutta steps allowed before a trajectory is declared failed.

FSTEP is the total number of Runge-Kutta steps allowed before the run is terminated.

Simple high rigidity trajectories often require only several hundred steps. Simple trajectories above the upper cutoff rigidity often can be completed in a few thousand steps. Most cosmic ray trajectories will complete in about 10,000 steps. Some quasi-trapped periodic orbits may require more than 100,000 steps. Trapped orbits require an infinite number of steps. Very low rigidity trajectories initiated at high polar latitudes will exhibit the quasi-trapped behavior and probably fail to reach a solution. (The step size criteria is based on the time to travel about one percent of a gyro-distance. Therefore trajectories with many loops require many steps to complete.)

Assuming the user wants to operate in a "batch mode" some job control parameters are needed. This is the quantity FSTEP. Some estimate of the computer speed is necessary. For desktop personal computers this can range from a few hundred steps per second on old obsolete 486 chips to the order of 50,000 steps per second obtainable with current Pentium® III chips operating at approximately 1 GHz clock cycle time. We have found a very significant difference in the program computational speed on the same computer that can be attributed to the efficiency of the object code generated by the compiler. In our testing on desktop platforms we have found that the executable code generated by the COMPAQ® Visual Fortran operates efficiently on a Microsoft® Windows operating system. The worst performing executable code (derived from an old, no longer sold system) ran about five times slower on the same test set of trajectory calculation. It is assumed that workstations will have trajectory computational speeds of the order or at least 10,000 steps per second. The default FSTEP setting will allow a batch run of the order of 10 hours if the program executes at 10,000 Runge-Kutta steps per second.

Program Operation

This program operates in the r, θ, ϕ coordinate system. The variables Y(1), Y(2), and Y(3) are the position vectors in the r, θ, ϕ coordinate system and the variables Y(4), Y(5), and Y(6) are the velocity vectors.

The program initially defines the physical constants used in the calculation and control parameters. It then enters a control loop beginning with reading a data line to determine the initial position and direction, the specified starting rigidity and how many trajectories to calculate at specified increments.

For each control line read, a call to subroutine GDGC converts the initial geodetic coordinates (map makers coordinates on the earth's surface) to geocentric r, θ, ϕ coordinates. Then the trajectory calculations are done by subroutine SINGLTJ

The control loop continues (read in control line, convert coordinates, do trajectory calculations) until a negative (or zero) value of rigidity is read in. When this occurs the program terminates.

Labeled Common arguments:

Block name:	/WRKVLU/
Arguments in block	F(6), Y(6), ERAD, EOMC, VEL, BR, BT, BP
F(6)	Array of force and acceleration vectors
Y(6)	Array of position and velocity vectors
ERAD	Average radius of the earth in kilometers

EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units)
VEL	Particle velocity in earth radii per second
BP	Value of the $B(\phi)$ magnetic field vector (in units of Gauss)
BR	Value of the $B(r)$ magnetic field vector (in units of Gauss)
BT	Value of the $B(\theta)$ magnetic field vector (in units of Gauss)
Block name: Arguments in block	/WRKTSC/ TSY2, TCY2, TSY3, TCY3
TSY2	Sine of the Y(2) coordinate (theta coordinate)
TCY2	Cosine of the Y(2) coordinate (theta coordinate)
TSY3	Sine of the Y(3) coordinate (phi coordinate)
TCY3	Cosine of the Y(3) coordinate (phi coordinate)
Block name: Arguments in block	/TRIG/ PI, RAD, PI02
PI	Value of pi
RAD	Value of degrees in a radian
PI02	Value of pi/2.0
Block name: Arguments in block	/GEOID/ ERADPL, ERECSQ
ERADPL	Polar radius of the earth in kilometers
ERECSQ	Eccentricity of ellipsoid squared
Block name: Arguments in block	/SNGLR/ SALT, DISOUT, GCLATD, GDLATD, GLOND, GDAZD, GDZED, RY1, RY2, RY3, RHT, TSTEP
SALT	Start altitude of trajectory above surface of geoid
DISOUT	Radial distance (in earth radii) for termination of calculation
GCLATD	Geocentric latitude in degree
GDLATD	Geodetic latitude in degrees
GLOND	East longitude in degrees
GDAZD	Geodetic azimuth in degrees
GDZED	Geodetic zenith in degrees
RY1	Original start position Y(1), (radial component in the r, θ, ϕ coordinate system)
RY2	Original start position Y(2), (theta component in the r, θ, ϕ coordinate system)
RY3	Original start position Y(3), (phi component in the r, θ, ϕ coordinate system)
RHT	Height above geoid where trajectory re-enters the atmosphere
TSTEP	Total number of steps in this run.
Block name: Arguments in block	/SNGLI/ LIMIT, NTRAJC, IERRPT
LIMIT	Maximum number of steps before 'failed' trajectory
NTRAJC	Number of trajectories calculated in this run
IERRPT	Integer control for printing diagnostics (normally = 0)

Subroutines called:

GCGC (TCD, TSD)

TCD	Cosine of the rotation angle
TSD	Sine of the rotation angle

SINGLTJ (PC, IRSLT, INDXPC, Y1GC, Y2GC, Y3GC)

PC	Particle rigidity (in units of GV)
IRSLT	Integer result of trajectory calculation (+1 = Allowed, 0 = Failed, -1 = Re-entrant)
INDXPC	Integer value of PC in units of MV
Y1GC	Y(1) position in geocentric coordinates
Y2GC	Y(2) position in geocentric coordinates
Y3GC	Y(3) position in geocentric coordinates

Dimensioned variables: all in labeled common

F(6)	Array of force and acceleration vectors
Y(6)	Array of position and velocity vectors

Data files: noneOutput files:

TAPE7 from subroutine SINGTJ (80 character summary)
 TAPE8 from subroutine SINGTJ (132 character line printer summary)
 TAPE16 (diagnostic output; if desired, set IERRPT to > 0)

Listing of all variables in program TJI95

B	Magnitude of the magnetic field (in Gauss) [in COMMON /WRKVLU/]
BP	Value of the $B(\phi)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BR	Value of the $B(r)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BT	Value of the $B(\theta)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
CF	Character variable "F"
DELPC	Increments of rigidity spacing search in control line
DISOUT	Distance (in earth radii) for trajectory termination [in COMMON /SNGLR/]
ERAD	Average radius of the earth in kilometers [in COMMON /WRKVLU/]
ERADPL	Polar radius of the earth in km [in COMMON /GEOID/]
ERECSQ	Eccentricity of ellipsoid squared [in COMMON /GEOID/]
F(6)	Array of "F" values (velocity and acceleration in program coordinates) [in COMMON /WRKVLU/]
GCLATD	Geocentric latitude in degrees (+ = north, south = -) [in COMMON /SNGLR/]
GDAZ	Geodetic azimuth in radians (measured clockwise from north)
GDAZD	Geodetic azimuth in degrees (measured clockwise from north)
GDLATD	Geodetic latitude in degrees (+ = north, south = -) [in COMMON /SNGLR/]
GDZE	Geodetic zenith in radians (0 = vertical)

GDZED	Geodetic zenith in degrees (0 = vertical)
GLOND	Geodetic East longitude in degrees (from Greenwich meridian) [in COMMON /SNGLR/]
IDE LPC	Integer value of rigidity change increment in MV (attempt to avoid round off)
IERRPT	Integer control for printing diagnostics (normally set to 0)
INDEX	Arbitrary index number of input control line (optional)
INDO	Integer control of number of trajectories to calculate
INDXPC	Integer value of rigidity in MV increments (attempt to avoid round off)
IOSTAT	Integer system argument of status of read
IRSLT	Internal result of particle trajectory (+1 = allowed, 0 = failed, -1 = re-entrant)
ISALT	Integer value of start altitude (in km) above geoid surface
LIMIT	Limit of number of steps before trajectory is declared "Failed"
LSTEP	Number of times step size control has been reduced to overcome trajectory error
NDO	Integer control read in (number of trajectories to compute from this control line)
NTRAJC	Number of trajectories in this computer run
PC	Rigidity of particle (in units of GV)
PI	Real value of the quantity Pi (~3.1415926535) [in COMMON /TRIG/]
PIO2	Real value of Pi divided by 2.0 [in COMMON /TRIG/]
RAD	Real value of one radian (~57.29578 degrees) [in COMMON /TRIG/]
RHT	Height above geoid where a trajectory re-entered the atmosphere
RY1	Real value of the starting position of the r coordinate in r, theta, phi coordinates [in COMMON /SNGLR/]
RY2	Real value of the starting position of the theta coordinate in r, theta, phi coordinates [in COMMON /SNGLR/]
RY3	Real value of the starting position of the phi coordinate in r, theta, phi coordinates [in COMMON /SNGLR/]
SALT	Real value of the starting altitude above the surface of the geoid [in COMMON /SNGLR/]
TCD	Trigonometric cosine of the rotation angle from geodetic to geocentric
TCGDAZ	Trigonometric cosine of the geodetic azimuth (Measured clockwise from north) [in COMMON /SNGLR/]
TCGDZE	Trigonometric cosine of the geodetic zenith (measured clockwise from north)
TCY2	Trigonometric cosine of the vector theta angle in r, theta, phi coordinates [in COMMON /SNGLR/]
TCY3	Trigonometric cosine of the vector phi angle in r, theta, phi coordinates [in COMMON /SNGLR/]
TSTEP	Number steps executed in this run
TSY2	Trigonometric sine of the vector theta (theta) angle in r, theta, phi coordinates [in COMMON /SNGLR/]
TSY3	Value of the trigonometric sine of the vector phi (phi) angle in r, theta, phi coordinates [in COMMON /SNGLR/]
VEL	Particle velocity in earth radii per second [in COMMON /WRKVLU/]

Y(6)	Array of "Y" values (position and velocity in r, θ, ϕ coordinates) [in COMMON /WRKVLU/]
Y1GC	Starting position r component in geocentric coordinates
Y1GD	Starting position r component in geodetic coordinates
Y2GC	Starting position θ component in geocentric coordinates
Y2GD	Starting position θ component in geodetic coordinates
Y3GC	Starting position ϕ component in geocentric coordinates
Y3GD	Starting position ϕ component in geodetic coordinates

Subroutine GDGC (TCD, TSD)

This subroutine calculates the angle between geodetic and geocentric coordinates. The arguments TCD and TSD are the trigonometric cosine and sine of the rotation angle from a normal from the surface of the geoid (geodetic coordinates) and a radial from the center of the earth (geocentric coordinates). See Appendix B of NSSDC 72-12)

Arguments in call statement

TCD Cosine of the rotation angle
 TSD Sine of the rotation angle

Labeled Common arguments:

Block name:	/WRKVLU/
Arguments in block	F(6), Y(6), ERAD, EOMC, VEL, BR, BT, BP
F(6)	Array of force and acceleration vectors
Y(6)	Array of position and velocity vectors
ERAD	Average radius of the earth in kilometers
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units)
VEL	Particle velocity in earth radii per second
BP	Value of the $B(\phi)$ magnetic field vector (in units of Gauss)
BR	Value of the $B(r)$ magnetic field vector (in units of Gauss)
BT	Value of the $B(\theta)$ magnetic field vector (in units of Gauss)
Block name:	/WRKTSC/
Arguments in block	TSY2, TCY2, TSY3, TCY3
TSY2	Sine of the $Y(2)$ coordinate (theta coordinate)
TCY2	Cosine of the $Y(2)$ coordinate (theta coordinate)
TSY3	Sine of the $Y(3)$ coordinate (phi coordinate)
TCY3	Cosine of the $Y(3)$ coordinate (phi coordinate)
Block name:	/TRIG/
Arguments in block	PI, RAD, PI02
PI	Value of pi
RAD	Value of degrees in a radian
PI02	Value of $\pi/2.0$
Block name:	/GEOID/
Arguments in block	ERADPL, ERECSQ
ERADPL	Polar radius of the earth in kilometers
ERECSQ	Eccentricity of ellipsoid squared
Block name:	/SNGLR/
Arguments in block	SALT, DISOUT, GCLATD, GDLATD, GLOND, GDAZD, GDZED, RY1, RY2, RY3, RHT, TSTEP
SALT	Start altitude of trajectory above surface of geoid
DISOUT	Radial distance (in earth radii) for termination of calculation
GCLATD	Geocentric latitude in degrees
GDLATD	Geodetic latitude in degrees
GLOND	East longitude in degrees

GDAZD	Geodetic azimuth in degrees
GDZED	Geodetic zenith in degrees
RY1	Original start position Y(1)
RY2	Original start position Y(2)
RY3	Original start position Y(3)
RHT	Height above geoid where trajectory re-enters the atmosphere
TSTEP	Total number of steps in this run.
Block name:	/SNGLI/
Arguments in block	LIMIT, NTRAJC, IERRPT
LIMIT	Maximum number of steps before 'failed' trajectory
NTRAJC	Number of trajectories calculated in this run
IERRPT	Integer control for printing diagnostics (normally = 0)
<u>Dimensioned variables:</u>	all in labeled common
F(6)	Array of force and acceleration vectors
Y(6)	Array of position and velocity vectors

Subroutines called: none

Data files: none

Output files: none

Operation:

The shape of the earth used is not a sphere, but an ellipsoid having a specified polar radius, equatorial radius, and eccentricity. When this subroutine is called, it defines the shape of an oblate earth from the polar and equatorial radius, and calculates vectors from a normal on the surface of the ellipsoid to the specified position in geodetic coordinates, at the specified latitude, and determines the vector rotation angle between geodetic coordinates and geocentric coordinates. The sine and cosine of this rotation angle are passed to the calling program. Geodetic latitude is a measure of latitude in a coordinate system normal to the surface of the earth. At a position on or above the surface of the ellipsoid, there is a slight difference between a direction normal to the surface of the ellipsoid and a direction to the geocentric. This difference is latitude dependent. (It is zero at the equator or poles and can be as large as approximately 1/2 of a degree at mid latitudes.) The vector rotation angle allows for direction specification in both geodetic (map) coordinates and geocentric coordinates. This small correction for the direction may be insignificant for some applications, but may be significant for precision calculation in a specific direction at high rigidities.

Data checking: none. The data to describe the shape of the earth are included in the subroutine.

Listing of all variables used in subroutine GDGC of program TJI95

B	Magnitude of the magnetic field (in Gauss) [in COMMON /WRKVLU/]
BP	Value of the $B(\phi)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]

BR	Value of the $B(r)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BT	Value of the $B(\theta)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
DISOUT	Distance (in earth radii) for trajectory termination [in COMMON /SNGLR/]
DISTKM	Starting position geocentric radial distance from geocenter
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units) [in COMMON /WRKVLU/]
ERAD	Average radius of the earth in kilometers [in COMMON /WRKVLU/]
ERADPL	Polar radius of the earth in km [in COMMON /GEOID/]
ERECSQ	Eccentricity of ellipsoid squared [in COMMON /GEOID/]
ERPLSQ	Polar radius of earth (in km) squared
F(6)	Array of "F" values (velocity and acceleration in program coordinates) [in COMMON /WRKVLU/]
GCLATD	Geocentric latitude in degrees (+ = north, south = -) [in COMMON /SNGLR/]
GDAZD	Geodetic azimuth in degrees (measured clockwise from north)
GDCLT	Geodetic co-latitude (in radians)
GDLATD	Geodetic latitude in degrees (+ = north, south = -) [in COMMON /SNGLR/]
GDZED	Geodetic zenith in degrees (0 = vertical)
GLOND	Geodetic East longitude in degrees (from Greenwich meridian) [in COMMON /SNGLR/]
ONE	Intermediate term in computations (see NSSDC ALLMAG description)
PI	Real value of the quantity Pi (~3.1415926535) [in COMMON /TRIG/]
PIO2	Real value of Pi divided by 2.0 [in COMMON /TRIG/]
RAD	Real value of one radian (~57.29578 degrees) [in COMMON /TRIG/]
RHO	Intermediate term in computations (see NSSDC ALLMAG description)
RHT	Height above geoid where a trajectory is declared re-entrant [in COMMON /SNGLR/]
RY1	Real value of the starting position of the r coordinate in r, θ, ϕ coordinates [in COMMON /SNGLR/]
RY2	Real value of the starting position of the theta coordinate in r, θ, ϕ coordinates [in COMMON /SNGLR/]
RY3	Real value of the starting position of the phi coordinate in r, θ, ϕ coordinates [in COMMON /SNGLR/]
SALT	Real value of the starting altitude above the surface of the geoid [in COMMON /SNGLR/]
TCD	Trigonometric cosine of the rotation angle from geodetic to geocentric
TCGDCLT	Trigonometric cosine of the geocentric co-latitude
TCY2	Trigonometric cosine of the vector theta angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TCY3	Trigonometric cosine of the vector phi angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
THREE	Intermediate term in computations (see NSSDC ALLMAG description)

TSD	Trigonometric sine of the rotation angle from geodetic to geocentric
TSGDCLT	Trigonometric sine of the geocentric co-latitude
TSTEP	Number steps executed in this run
TSY2	Trigonometric sine of the vector theta (θ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TSY3	Trigonometric sine of the vector phi (ϕ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TWO	Intermediate term in computations (see NSSDC ALLMAG description)
VEL	Particle velocity in earth radii per second [in COMMON /WRKVLU/]
Y(6)	Array of "Y" values (position and velocity in r, θ, ϕ coordinates) [in COMMON /WRKVLU/]

Reference publication:

ALLMAG, GCAZLMG, LINTRA: Commuter Programs For Geomagnetic Field And Field-Line Calculations, E.G. Stassinopoulos and G.D. Mead, NSSDC 72-12, February 1972, NASA, GFSC, Greenbelt MD.

Subroutine SINGLTJ (PC, IRSLT, INDXPC, Y1GC, Y2GC, Y3GC)

This subroutine does the actual trajectory tracing. When called it initially defines control parameters and constants used in the particle tracing and initializes the Runge-Kutta variables to zero. It sets up the initial position and direction, and defines the relativistic parameters relating to the particle total energy and speed.

In this version of the subroutine, an oxygen nuclei (^{16}O) is used as the test particle. By definition a ^{16}O nuclei has a mass of 16 Atomic Mass Units (AMU) and an atomic charge of 8. The mass-energy conversion for one AMU is 0.93114 GeV. If it were desired to modify the program for some other nuclei, such as a proton that has an atomic charge of 1 and atomic mass of 1.0081415 AMU, then the rest mass energy for atomic nuclei must be adjusted.

After the initial definitions, the subroutine then chooses an initial starting step length (a relatively small value) and starts the Runge-Kutta process of tracing the particle trajectory. After each step it goes though an error checking and detection process. If the checks are satisfactory, it determines the particle location with respect to the atmosphere and the outer boundary.

If the charged particle is between the atmosphere and the outer boundary, it adjusts the size of the next step and continues the trajectory tracing until the LIMIT on the number of steps is reached.

If the charged particle is entering the atmosphere, it terminates the calculation.

If the charged particle is less than 100 km above the earth's surface, it maintains a running sum of the time at low altitudes.

If the charged particle is approaching the outer boundary, it adjusts the step size so it penetrates this boundary at small step lengths.

If the charged particle has penetrated the outer boundary at a small step, it computes the final coordinates.

When the particle has reached a solution (allowed or re-entrant) or reached the step limit, it writes out the result and returns to the calling program.

Arguments in call statement

PC, IRSLT, INDXPC, Y1GC, Y2GC, Y3GC

PC	Particle rigidity
IRSLT	Integer result of trajectory (+1 = allowed, 0 = failed, -1 = re-entrant)
INDXPC	Integer value of PC in MV units
Y1GC	Y(1) position in geocentric coordinates
Y2GC	Y(2) position in geocentric coordinates
Y3GC	Y(3) position in geocentric coordinates

Labeled Common arguments:

Block name:	/WRKVLU/
Arguments in block	F(6), Y(6), ERAD, EOMC, VEL, BR, BT, BP
F(6)	Array of force and acceleration vectors
Y(6)	Array of position and velocity vectors
ERAD	Average radius of the earth in kilometers
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units)
VEL	Particle velocity in earth radii per second
BP	Value of the $B(\phi)$ magnetic field vector (in units of Gauss)
BR	Value of the $B(r)$ magnetic field vector (in units of Gauss)

BT	Value of the $B(\theta)$ magnetic field vector (in units of Gauss)
Block name:	/WRKTSC/
Arguments in block	TSY2, TCY2, TSY3, TCY3
TSY2	Sine of the Y(2) coordinate (theta coordinate)
TCY2	Cosine of the Y(2) coordinate (theta coordinate)
TSY3	Sine of the Y(3) coordinate (phi coordinate)
TCY3	Cosine of the Y(3) coordinate (phi coordinate)
Block name:	/TRIG/
Arguments in block	PI, RAD, PI02
PI	Value of pi
RAD	Value of degrees in a radian
PI02	Value of pi/2.0
Block name:	/GEOID/
Arguments in block	ERADPL, ERECSQ
ERADPL	polar radius of the earth in kilometers
ERECSQ	Eccentricity of ellipsoid squared
Block name:	/SNGLR/
Arguments in block	SALT, DISOUT, GCLATD, GDLATD, GLOND, GDAZD, GDZED, RY1, RY2, RY3, RHT, TSTEP
SALT	Start altitude of trajectory above surface of geoid
DISOUT	Radial distance (in earth radii) for termination of calculation
GCLATD	Geocentric latitude in degrees
GDLATD	Geodetic latitude in degrees
GLOND	East longitude in degrees
GDAZD	Geodetic azimuth in degrees
GDZED	Geodetic zenith in degrees
RY1	Original start position Y(1), (radial component in the r, θ, φ coordinate system)
RY2	Original start position Y(2), (theta component in the r, θ, φ coordinate system)
RY3	Original start position Y(3), (phi component in the r, θ, φ coordinate system)
RHT	Height above geoid where trajectory re-enters the atmosphere
TSTEP	Total number of steps in this run.
Block name:	/SNGLI/
Arguments in block	LIMIT, NTRAJC, IERRPT
LIMIT	Maximum number of steps before 'failed' trajectory
NTRAJC	Number of trajectories calculated in this run
IERRPT	Integer control for printing diagnostics (normally = 0)
<u>Dimension variables:</u> (not in labeled common)	
P(6), Q(6), R(6), S(6), YB(6), FOLD(6), YOLD(6)	
P(6)	Runge-Kutta variable
Q(6)	Runge-Kutta variable
R(6)	Runge-Kutta variable
S(6)	Runge-Kutta variable
YB(6)	Runge-Kutta variable
FOLD(6)	"F" vectors of previous step
YOLD(6)	"Y" vectors of previous step

Subroutines called: FGRADData files: noneOutput files:

TAPE7 (80 character summary)
TAPE8 (132 character line printer summary)
TAPE16 (diagnostic output; if desired set IERRPT to > 0)

Program Operation:

This program operates in the r, θ, ϕ coordinate system. The variables $Y(1), Y(2)$, and $Y(3)$ are the position vectors in the r, θ, ϕ coordinate system and the variables $Y(4), Y(5)$, and $Y(6)$ are the velocity vectors.

When this subroutine is called, it initially defines control parameters and constants used in the particle path tracing, and initializes the Runge-Kutta variables to zero. It obtains the particle's height with respect to the surface of an oblate earth. It sets up the initial position vectors, $Y(1), Y(2)$ and $Y(3)$, and based on the particle rigidity, sets up velocity vectors, $Y(4), Y(5)$, and $Y(6)$. It then defines the relativistic parameters TENG (total energy), EOMC (charge per relativistic mass/energy equivalent), and GMA (the relativistic parameter of total energy over the rest mass energy). It defines scalar quantities relating to the particle, BETA (the particle speed with respect to light), PVEL (the particle speed in earth radii per second), and HMAX (a maximum step length allowed for this particle rigidity).

Next it defines an initial starting step length (a relatively small value) and starts the Runge-Kutta process of tracing the particle trajectory. Comment cards specifically indicate the Runge-Kutta iteration process, which is the coding between FORTRAN statement numbers 130 and 170. The calls to subroutine FGRAD evaluate the $V \times B$ force on the particle during this step. The logic is very similar to that documented in Ralston and Wilf (1960). After each Runge-Kutta iteration step there is an extensive error checking and detection process.

The error checking process begins with a check on the particle speed (BETA), which should remain invariant throughout the trajectory. If the difference between the initial particle speed (BETA) and its current speed (RCKBETA) is greater than EDIF, then the trajectory tracing process is re-initialized (including the NSTEP variable) and the trajectory re-started at a smaller step size selection criteria. Up to five re-starts are allowed before the specific trajectory is declared impossible to calculate, evaluated as "failed", and the path length made negative in order to distinguish it from successful trajectories. In order to attempt to reach a solution the EDIF variable is widened by a factor of two after each successive trajectory failure.

After the error check, then the acceleration of the particle is compared with previous values. We have found that computational errors are most likely to occur when there are rapid changes in the acceleration. If the average change in acceleration exceeds a factor of five, or if any component of the acceleration exceeds a factor of three, then the step length for the next Runge-Kutta step is reduced.

Along the particle path the software checks the particle location with respect to the atmosphere and the outer boundary. If the charged particle is less than 100 km above the earth's surface, it maintains a running sum of the time at low altitudes. If the charged particle is entering the atmosphere, it terminates the calculation.

The next check determines if the particle has penetrated the outer termination boundary. The step length can be relatively large at extreme distances from the earth. If the outer boundary has been penetrated at a large step size, the trajectory is "backed up" and the step size reduced until it penetrates the boundary at a small step size. This results in a more precise determination of the penetration location and can significantly affect the computed asymptotic direction.

If there are no errors and the charged particle is between the atmosphere and the outer boundary, the software adjusts the size of the next step appropriate for the magnitude of the magnetic field (the step size is normally about one percent of the gyro-distance) and continues the trajectory tracing. The basic step length algorithm is:

$$H = ((2.0 * \pi * 33.333 * PC) / (B * \beta * C)) / 100.0$$

where "H" is time in seconds, "PC" is the particle rigidity in GV, "B" is the magnitude of the magnetic field in Gauss and "C" is the speed of light in km/sec.

(A handy formula to remember is the gyro-radius is 33 km per GV per Gauss)

The software initially starts at a trajectory calculation at a small step size and the step size is permitted to grow at a maximum of about 20 percent each step. If the particle trajectory starts to loop to a lower altitude, then the step size is reduced to compensate for the increasing magnitude of the magnetic field. When a loop in the trajectory develops, the acceleration forces increase and step size adjustments are made in the case of a significant increase in the acceleration forces.

During the trajectory tracing, the software notes the number of maximum and minima the trajectory experiences. This information is useful in ascertaining the complexity of the trajectory.

When the particle has reached a solution (allowed or re-entrant) or reached the step limit, the subroutine will write out the result and return to the calling program. The fate is coded in the variable IFATE: (0 = Allowed, 1 = Failed, 2 = Re-entrant, 3 = Failed, but max alt > 6.6 earth radii)

If the trajectory is allowed (penetrates the outer boundary), then the velocity vectors are transformed into asymptotic latitude and longitude. Asymptotic latitude and longitude are the geocentric coordinates the velocity vector would have at infinity. If the trajectory re-enters the atmosphere, then the position coordinates are transformed to geocentric latitude and longitude.

The 'output' of the software is in summary files, on one line for each trajectory calculation.

The output files are:

TAPE7 is in an 80-column "card image" format. This contains the initial conditions (the geodetic latitude, longitude, rigidity, zenith, and azimuth), the final results (asymptotic latitude and longitude, fate and number of steps), and a magnetic field identifier.

TAPE8 is in a line printer 132-column format. This contains more detail; the initial conditions (the geodetic latitude and the geocentric latitude, the longitude, rigidity, zenith, and azimuth), the final results (asymptotic latitude and longitude, path length,

trajectory time, time at altitudes under 100 km, number of maximum and minimum in trajectory, fate, and number of steps), and a magnetic field identifier.

TAPE16 is a diagnostic output, including a record of restarts due to BETA checks or trajectory failures.

See the examples section for samples of the output.

Possible Additions for Trajectory Plotting

If it is desired to plot a trajectory, the position variables Y(1), Y(2), and Y(3) must be stored after each Runge-Kutta step in a suitable array. The task of adding such a modification should be straightforward and is left to the individual program user.

Listing of all variables in subroutine SINGLTJ

ACCR	Magnitude of current value of the particle acceleration
ACCOLD	Magnitude of last value of the particle acceleration
AFOLD	Absolute value of FOLD
AHLT	Variable to control step size at high latitude
ANUC	Atomic number of number of nucleons in atom
ATRG1	Intermediate value to compute asymptotic latitude (avoid 0/0 problem)
ATRG2	Intermediate value to compute asymptotic latitude (avoid 0/0 problem)
AZD	Azimuth angle in degrees (measured clockwise from north)
B	Magnitude of the magnetic field (in Gauss) [in COMMON /WRKVLU/]
BETA	Particle speed as fraction of light speed ($\beta = v/c$ where c is speed of light)
BETAST	Control variable for reducing step size if error has occurred
BP	Value of the $B(\phi)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BR	Value of the $B(r)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BT	Value of the $B(\theta)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
CF	Character variable "F"
CNAME	Character variable (up to 6 characters) identifying magnetic field used
CR	Character variable "R"
DELACC	Change in particle acceleration from previous step
DISCK	Step length control variable to approach boundary at 10 % increments
DISOUT	Distance (in earth radii) for trajectory termination [in COMMON /SNGLR/]
DISTR	Radial distance (in earth radii) from current distance to termination boundary
EDIF	Variation in β allowed before error declared
EMCSQ	Mass energy equivalent for a AMU (0.931141 GeV)
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units) [in COMMON /WRKVLU/]
ERAD	Average radius of the earth in kilometers [in COMMON /WRKVLU/]
ERADPL	Polar radius of the earth in km [in COMMON /GEOID/]
ERECSQ	Eccentricity of ellipsoid squared [in COMMON /GEOID/]
F(6)	Array of "F" values (velocity and acceleration in program coordinates)

	[in COMMON /WRKVLU/]
FOLD(6)	Array of "F" values (velocity and acceleration) from previous step
FASLAT	Asymptotic latitude (in degrees)
FASLON	Asymptotic longitude (in degrees east of the Greenwich meridian)
GCLATD	Geocentric latitude in degrees (+ = north, south = -) [in COMMON /SNGLR/]
GDAZD	Geodetic azimuth in degrees (measured clockwise from north)
GDLATD	Geodetic latitude in degrees (+ = north, south = -) [in COMMON /SNGLR/]
GDZED	Geodetic zenith in degrees (0 = vertical)
GLOND	Geodetic East longitude in degrees (from Greenwich meridian)
	[in COMMON /SNGLR/]
GMA	Relativistic factor (total energy/rest energy)
GRNDKM	Altitude above surface of earth at this latitude (in km)
H	Runge-Kutta step size (in seconds)
HB	Preliminary value of step size for $\beta = 1$
HCK	Control to limit step size growth to 20%
HCNG	Change of step size from previous step
HMAX	Maximum value of step size allowed
HOLD	Value of previous Runge-Kutta step size
HSNEK	Control to approach 90% of distance to boundary
HSTART	Starting step size value (deliberately made small)
I	Index variable in do loops
IAZ	Integer value of azimuth (measured counter clockwise from north)
ICK	Index for checking acceleration growth
IERRPT	Integer control for printing diagnostics (normally set to 0)
IFATE	Integer fate of particle trajectory (0 = Allowed, 1 = Failed, 2 = Re-entrant 3 = Failed, but max alt > 6.6 earth radii)
INDXPC	Index of particle rigidity in MV
IRT	Integer control for writing results (+1 = Allowed, 0 = Failed, -1 = Re-entrant)
IRSLT	Internal result of particle trajectory (+1 = Allowed, 0 = Failed, -1 = Re-entrant)
ISALT	Integer value of start altitude (in km) above earth surface
IZE	Integer value of zenith angle (in degrees)
KBF	Number of failed attempts to trace this trajectory
LIMIT	Limit of number of steps before trajectory declared "Failed"
LSTEP	Number of times the step size control reduced to overcome trajectory error
NMAX	Number of maxima in complex trajectory path
NMIN	Number of minima in complex trajectory path
NSTEP	Number of steps in current trajectory
NSTEPT	Temporary variable that can be used to print out first 1000 steps
NTRAJC	Number of trajectories in this computer run
P(6)	Array of intermediate values used in Runge-Kutta integration
PATH	Total distance of trajectory path from start to termination
PC	Rigidity of particle (in units of GV)
PI	Real value of the quantity Pi (~3.1415926535) [in COMMON /TRIG/]
PIO2	Real value of Pi divided by 2.0 [in COMMON /TRIG/]

PSALT	Current particle distance from ground (used for re-entrant calculations)
PTCY2	Absolute value of cosine Y(2) (used in control of polar step size)
PVEL	Particle velocity (in earth radii per second)
R(6)	Array of intermediate values used in Runge-Kutta integration
R100KM	Y(1) distance of 100 km altitude at this latitude
R120KM	Y(1) distance of 120 km altitude at this latitude
RAD	Real value of one radian (~57.29578 degrees) [in COMMON /TRIG/]
RC1O6	Constant in Runge-Kutta integration (1.0/6.0)
RCKBETA	Current value of particle β after this step
RENLAT	Latitude of re-entrant particle intersection with atmosphere
RENLON	Longitude of re-entrant particle intersection with atmosphere
RFA	Ratio of acceleration magnitude between current step and last step
RFCK	Ratio of acceleration component between current step and last step
RHT	Height above geoid where a trajectory re-entered the atmosphere
RY1	Real value of the starting position of the r coordinate in r, θ, ϕ coordinates [in COMMON /SNGLR/]
RY2	Real value of the starting position of the theta coordinate in r, θ, ϕ coordinates [in COMMON /SNGLR/]
RY3	Real value of the starting position of the phi coordinate in r, θ, ϕ coordinates [in COMMON /SNGLR/]
S(6)	Array of intermediate values used in Runge-Kutta integration
SALT	Real value of the starting altitude above the surface of the geoid [in COMMON /SNGLR/]
SR2	Runge-Kutta constant (square root of 2.0)
TAU	Time (in seconds) for a trajectory transit from start to termination
TBETA	Difference between current value of β and starting value of β
TCY2	Trigonometric cosine of the vector theta (θ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TCY3	Trigonometric cosine of the vector phi (ϕ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TENG	Total energy of particle (kinetic energy plus rest mass energy)
TMS2O2	Runge-Kutta constant (2.0 - SR2/2.0)
TPS2O2	Runge-Kutta constant ((2.0 + SR2/2.0))
TSTEP	Number steps executed in this run
TSY2	Trigonometric sine of the vector theta (θ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TSY2SQ	Square of TSY2
TSY3	Value of the trigonometric sine of the vector phi (ϕ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TU100	Time (in seconds) the particle is under 100 km altitude
VEL	Particle velocity in earth radii per second [in COMMON /WRKVLU/]
Y(6)	Array of "Y" values (position and velocity in r, θ, ϕ coordinates) [in COMMON /WRKVLU/]
YB(6)	Array of intermediate values used in Runge-Kutta integration
Y10	Y(1) radial coordinate for re-entrant distance at this latitude

Y1GC	Starting position r component in geocentric coordinates
Y2GC	Starting position θ component in geocentric coordinates
Y3GC	Starting position ϕ component in geocentric coordinates
YDA5	Intermediate value for computing asymptotic latitude
YMAX	Maximum radial distance attained by trajectory
ZCHARGE	Atomic charge number
ZED	Zenith angle in degrees

Reference publication:

Ralston, A, and Wilf, S.H., Mathematical Models for Digital Computers, John Wiley and Sons, New York, 1960.

Subroutine FGRAD

This subroutine calculates the $\mathbf{V} \times \mathbf{B}$ force on the charged particle.

Arguments in call statement: none. All in labeled common

Labeled Common arguments:

Block name:	/WRKVLU/
Arguments in block	F(6), Y(6), ERAD, EOMC, VEL, BR, BT, BP
F(6)	Array of force and acceleration vectors
Y(6)	Array of position and velocity vectors
ERAD	Average radius of the earth in kilometers
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units)
VEL	Particle velocity in earth radii per second
BP	Value of the $B(\phi)$ magnetic field vector (in units of Gauss)
BR	Value of the $B(r)$ magnetic field vector (in units of Gauss)
BT	Value of the $B(\theta)$ magnetic field vector (in units of Gauss)
Block name:	/WRKTSC/
Arguments in block	TSY2, TCY2, TSY3, TCY3
TSY2	Sine of the $Y(2)$ coordinate (theta coordinate)
TCY2	Cosine of the $Y(2)$ coordinate (theta coordinate)
TSY3	Sine of the $Y(3)$ coordinate (phi coordinate)
TCY3	Cosine of the $Y(3)$ coordinate (phi coordinate)

Subroutines called:

MAGNEW95

(All arguments are in labeled common /WRKVLU/ and /WRKTSC/)

Data files: none

Output files: none

Program Operation:

When this subroutine is called, the force vectors, ((F(1), F(2), F(3))) are defined. The sine and cosine of the $Y(2)$ coordinates are determined. The magnetic field vectors (BR, BT, BP) at the particle position ($Y(1)$, $Y(2)$, $Y(3)$) are obtained by the call to subroutine MAGNEW95. Then the acceleration vectors (F(4), F(5), F(6)) are calculated.

Listing of all variables in subroutine FGRAD in program TJI95

B	Magnitude of the magnetic field (in Gauss) [in COMMON /WRKVLU/]
BP	Value of the $B(\phi)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]

BR	Value of the B(r) magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BT	Value of the B(θ) magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units) [in COMMON /WRKVLU/]
ERAD	Average radius of the earth in kilometers [in COMMON /WRKVLU/]
F(6)	Array of "F" values (velocity and acceleration in program coordinates) [in COMMON /WRKVLU/]
SQY6	Intermediate term $(Y_6)^* Y_6 / Y_1$
TAY2	Intermediate term $T\sin\theta \cos\phi$
TCY2	Trigonometric cosine of the vector theta angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TCY3	Trigonometric cosine of the vector phi angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TSY2	Trigonometric sine of the vector theta (θ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TSY3	Trigonometric sine of the vector phi (ϕ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
VEL	Particle velocity in earth radii per second [in COMMON /WRKVLU/]
Y(6)	Array of "Y" values (position and velocity in r, θ, ϕ coordinates) [in COMMON /WRKVLU/]
Y5OY1	Intermediate term $Y_5^* Y_1$

Subroutine MAGNEW95

This is the magnetic field evaluation program containing the IGRF95 magnetic field model. The normalized magnetic field coefficients have been pre-processed and loaded in as data statements. This is a serial computation designed to repeatedly evaluate the same magnetic field model at different positions in space at optimum efficiency. Each term of the magnetic field expansion is the derivative of the previous term.

Arguments in call statement: none. (All in labeled common)

Labeled Common arguments:

Block name:	/WRKVLU/
Arguments in block	F(6), Y(6), ERAD, EOMC, VEL, BR, BT, BP
F(6)	Array of force and acceleration vectors
Y(6)	Array of position and velocity vectors
ERAD	Average radius of the earth in kilometers
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units)
VEL	Particle velocity in earth radii per second
BP	Value of the $B(\phi)$ magnetic field vector (in units of Gauss)
BR	Value of the $B(r)$ magnetic field vector (in units of Gauss)
BT	Value of the $B(\theta)$ magnetic field vector (in units of Gauss)
Block name:	/WRKTSC/
Arguments in block	TSY2, TCY2, TSY3, TCY3
TSY2	Sine of the $Y(2)$ coordinate (theta coordinate)
TCY2	Cosine of the $Y(2)$ coordinate (theta coordinate)
TSY3	Sine of the $Y(3)$ coordinate (phi coordinate)
TCY3	Cosine of the $Y(3)$ coordinate (phi coordinate)

Dimensioned variables: (not in labeled common)

G(11,11), BM(11)	Normalized coefficients ordered for fast serial computation
BM(11)	Values to determine if the expansion should terminated at order N

Program Operation:

This subroutine is designed for efficient serial computation of the earth's main magnetic field. This procedure expands the terms into FORTRAN coding (resulting in pages and pages of FORTRAN code) and then evaluates the normalized field coefficients. The result of this serial expansion is approximately an order of magnitude speed increase over the recursion method which is much more compact in program size but requires the expansion of the Legender polynomials each time the subroutine is called.

When this subroutine is called it checks the value of the variable JDATA. If JDATA is not 77, then it loads in the data coefficients; otherwise it proceeds directly to the magnetic field evaluation. The array G(11,11) contains the pre-processed and normalized magnetic field model coefficients ordered for fast serial computation. The array BM(11) contains a set of values that determine if the expansion should be

terminated above a specified order because the additional contribution of the magnetic field would not be significant. (This is an additional technique to speed up the computation.)

At the beginning of the magnetic field determination, the sine and cosine of Y(3) (the phi coordinate) are calculated. (This has not been done since the completion of the last Runge-Kutta step.) It then determines the value of the AR variable; AR is reciprocal of the radial distance, (Y(1)) ,in earth radii. Each order of the field expansion requires an evaluation of AR^N where N is the order of the field expansion. (The N = 1 term, a description of a monopole magnetic field, is zero.) The first magnetic field evaluation designated by N = 2 is the dipole field component. Each subsequent order of expansion evaluates the contribution of the next order and adds this to the contribution of the previous orders. The final computation converts the magnetic field vectors to units of Gauss. (One Gauss - 10^5 Nt). A more detailed description of the process is given in NSSDC DATA USERS NOTE 68-11.

Listing of all variables in subroutine MAGNEW95

AOR	Temporary value of radial distance (in earth radii) to the N th power
AR	1.0/Radial distance (in earth radii) [AR = 1.0/Y(1)]
B	Magnitude of the magnetic field (in Gauss) [in COMMON /WRKVLU/]
BERR	Variable for dropping terms if field magnitude is less than significant value (set to 0.001 for improved accuracy)
BM(11)	Array of check values for dropping terms of magnetic field expansion
BP	Value of the B(ϕ) magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BR	Value of the B(r) magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BT	Value of the B(θ) magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
DP101	Derivative of the polynomial (10,1) term
DP102	Derivative of the polynomial (10,2) term
DP103	Derivative of the polynomial (10,3) term
DP104	Derivative of the polynomial (10,4) term
DP105	Derivative of the polynomial (10,5) term
DP106	Derivative of the polynomial (10,6) term
DP107	Derivative of the polynomial (10,7) term
DP108	Derivative of the polynomial (10,8) term
DP109	Derivative of the polynomial (10,9) term
DP111	Derivative of the polynomial (11,1) term
DP110	Derivative of the polynomial (11,10) term
DP111	Derivative of the polynomial (11,11) term
DP112	Derivative of the polynomial (11,2) term
DP113	Derivative of the polynomial (11,3) term
DP114	Derivative of the polynomial (11,4) term
DP115	Derivative of the polynomial (11,5) term
DP116	Derivative of the polynomial (11,6) term
DP117	Derivative of the polynomial (11,7) term
DP118	Derivative of the polynomial (11,8) term
DP119	Derivative of the polynomial (11,9) term
DP21	Derivative of the polynomial (2,1) term
DP22	Derivative of the polynomial (2,2) term
DP31	Derivative of the polynomial (3,1) term
DP32	Derivative of the polynomial (3,2) term
DP33	Derivative of the polynomial (3,3) term

DP41	Derivative of the polynomial (4,1) term
DP42	Derivative of the polynomial (4,2) term
DP43	Derivative of the polynomial (4,3) term
DP44	Derivative of the polynomial (4,4) term
DP51	Derivative of the polynomial (5,1) term
DP52	Derivative of the polynomial (5,2) term
DP53	Derivative of the polynomial (5,3) term
DP54	Derivative of the polynomial (5,4) term
DP55	Derivative of the polynomial (5,5) term
DP61	Derivative of the polynomial (6,1) term
DP62	Derivative of the polynomial (6,2) term
DP63	Derivative of the polynomial (6,3) term
DP64	Derivative of the polynomial (6,4) term
DP65	Derivative of the polynomial (6,5) term
DP66	Derivative of the polynomial (6,6) term
DP71	Derivative of the polynomial (7,1) term
DP72	Derivative of the polynomial (7,2) term
DP73	Derivative of the polynomial (7,3) term
DP74	Derivative of the polynomial (7,4) term
DP75	Derivative of the polynomial (7,5) term
DP76	Derivative of the polynomial (7,6) term
DP77	Derivative of the polynomial (7,7) term
DP81	Derivative of the polynomial (8,1) term
DP82	Derivative of the polynomial (8,2) term
DP83	Derivative of the polynomial (8,3) term
DP84	Derivative of the polynomial (8,4) term
DP85	Derivative of the polynomial (8,5) term
DP86	Derivative of the polynomial (8,6) term
DP87	Derivative of the polynomial (8,7) term
DP88	Derivative of the polynomial (8,8) term
DP91	Derivative of the polynomial (9,1) term
DP92	Derivative of the polynomial (9,2) term
DP93	Derivative of the polynomial (9,3) term
DP94	Derivative of the polynomial (9,4) term
DP95	Derivative of the polynomial (9,5) term
DP96	Derivative of the polynomial (9,6) term
DP97	Derivative of the polynomial (9,7) term
DP98	Derivative of the polynomial (9,8) term
DP99	Derivative of the polynomial (9,9) term
DP1010	Derivative of the polynomial (10,10) term
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units) (not used in this subroutine) [in COMMON /WRKVLU/]
ERAD	Average radius of the earth in kilometers (not used in this subroutine) [in COMMOM /WRKVLU/]
ERR	Intermediate term used to evaluate when higher order terms can be dropped
F(6)	Array of "F" values (velocity and acceleration in program coordinates) [in COMMON /WRKVLU/]
G(11,11)	Array of normalized magnetic field coefficients ordered for fast computation

GMSUM	Check value to assure that proper magnetic field coefficients are loaded
GSUM	Check data for testing that proper magnetic field coefficients are loaded
JDATA	Integer test value for loading in data statement for magnetic field coefficients
JMAG	Integer order of magnetic field expansion (N+1)
L	Intermediate index for data checking
M	Intermediate index for data checking
MGNMAX	Integer value of maximum order of magnetic field expansion
P101	Polynomial (10,1) term
P1010	Polynomial (10,10) term
P102	Polynomial (10,2) term
P103	Polynomial (10,3) term
P104	Polynomial (10,4) term
P105	Polynomial (10,5) term
P106	Polynomial (10,6) term
P107	Polynomial (10,7) term
P108	Polynomial (10,8) term
P109	Polynomial (10,9) term
P111	Polynomial (11,1) term
P1110	Polynomial (11,10) term
P1111	Polynomial (11,11) term
P112	Polynomial (11,2) term
P113	Polynomial (11,3) term
P114	Polynomial (11,4) term
P115	Polynomial (11,5) term
P116	Polynomial (11,6) term
P117	Polynomial (11,7) term
P118	Polynomial (11,8) term
P119	Polynomial (11,9) term
P21	Polynomial (2,1) term
P22	Polynomial (2,2) term
P31	Polynomial (3,1) term
P32	Polynomial (3,2) term
P33	Polynomial (3,3) term
P41	Polynomial (4,1) term
P42	Polynomial (4,2) term
P43	Polynomial (4,3) term
P44	Polynomial (4,4) term
P51	Polynomial (5,1) term
P52	Polynomial (5,2) term
P53	Polynomial (5,3) term
P54	Polynomial (5,4) term
P55	Polynomial (5,5) term
P61	Polynomial (6,1) term
P62	Polynomial (6,2) term
P63	Polynomial (6,3) term
P64	Polynomial (6,4) term
P65	Polynomial (6,5) term

P66	Polynomial (6,6) term
P71	Polynomial (7,1) term
P72	Polynomial (7,2) term
P73	Polynomial (7,3) term
P74	Polynomial (7,4) term
P75	Polynomial (7,5) term
P76	Polynomial (7,6) term
P77	Polynomial (7,7) term
P81	Polynomial (8,1) term
P82	Polynomial (8,2) term
P83	Polynomial (8,3) term
P84	Polynomial (8,4) term
P85	Polynomial (8,5) term
P86	Polynomial (8,6) term
P87	Polynomial (8,7) term
P88	Polynomial (8,8) term
P91	Polynomial (9,1) term
P92	Polynomial (9,2) term
P93	Polynomial (9,3) term
P94	Polynomial (9,4) term
P95	Polynomial (9,5) term
P96	Polynomial (9,6) term
P97	Polynomial (9,7) term
P98	Polynomial (9,8) term
P99	Polynomial (9,9) term
RC10	Intermediate value (N=10) in magnetic field expansion
RC11	Intermediate value (N=11) in magnetic field expansion
RC2	Intermediate value (N=2) in magnetic field expansion
RC3	Intermediate value (N=3) in magnetic field expansion
RC4	Intermediate value (N=4) in magnetic field expansion
RC5	Intermediate value (N=5) in magnetic field expansion
RC6	Intermediate value (N=6) in magnetic field expansion
RC7	Intermediate value (N=7) in magnetic field expansion
RC8	Intermediate value (N=8) in magnetic field expansion
RC9	Intermediate value (N=9) in magnetic field expansion
TCP10	Trigonometric cosine function for the P10 term
TCP11	Trigonometric cosine function for the P11 term
TCP2	Trigonometric cosine function for the P2 term
TCP3	Trigonometric cosine function for the P3 term
TCP4	Trigonometric cosine function for the P4 term
TCP5	Trigonometric cosine function for the P5 term
TCP6	Trigonometric cosine function for the P6 term
TCP7	Trigonometric cosine function for the P7 term
TCP8	Trigonometric cosine function for the P8 term
TCP9	Trigonometric cosine function for the P9 term
TCY2	Trigonometric cosine of the vector theta angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TCY3	Trigonometric cosine of the vector phi angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]

\equiv h radii per second (not used in this subroutine)
/WRKVLU/]

Reference publication:

Computation of the Main Geomagnetic Field position and velocity in r, θ, ϕ coordinates
NSSDC 68-11, February 1968, NSDC, NASA, /WRKVLU/]

Dr Spherical Harmonic Expansion, Data Users No
GFSC, Greenbelt MD.

Final Report Grant NAG5-8009, Section II, Part1, TJI951

TSP10	Trigonometric sine function for the P10 term
TSP11	Trigonometric sine function for the P11 term
TSP2	Trigonometric sine function for the P2 term
TSP3	Trigonometric sine function for the P3 term
TSP4	Trigonometric sine function for the P4 term
TSP5	Trigonometric sine function for the P5 term
TSP6	Trigonometric sine function for the P6 term
TSP7	Trigonometric sine function for the P7 term
TSP8	Trigonometric sine function for the P8 term
TSP9	Trigonometric sine function for the P9 term
TSY2	[in COMMON] Trigonometric sine function for the P5 term
TSY3	[in COMMON] Trigonometric sine function for the P6 term
	[in COMMON] Trigonometric sine function for the P7 term
	[in COMMON] Trigonometric sine function for the P8 term
	[in COMMON] Trigonometric sine function for the P9 term
VEL	Particle velocity in earth vector theta angle in r, θ, ϕ coordinates [in COMMON] /SNGLR/
Y(6)	Array of "Y" values (phi vector phi angle in r, θ, ϕ coordinates [in COMMON] /SNGLR/)

DESCRIPTION OF PROGRAM TJI95T

This software package is self-contained and capable of being compiled and executed on a variety of platforms ranging from a personal computer to large scale "super computers". The software is written in FORTRAN 77. The software is designed to efficiently compute the trajectory of an energetic charged particle of a specified momentum per unit charge (rigidity) through a model magnetic field. For cosmic ray access to the earth, the geocenter becomes the origin of the coordinate system. All calculations are done in the r, θ, ϕ coordinate system (a right-handed, orthogonal coordinate system). The magnetic field subroutine included in this software is designed for efficient evaluation of the IGRF95 model of the earth's magnetic field.

In its usual mode of computing the path of cosmic ray trajectories in a model of the earth's magnetic field, we utilize this program to determine the path of a cosmic ray (a positively charged particle) from interplanetary space arriving at the earth at a specified position and direction. To accomplish this, a negatively charged particle is 'launched' from the 'top' of the atmosphere at a specified position (latitude and longitude) in a specific direction (zenith and azimuth), and its path traced though the model magnetic field until it either (1) reaches a specified radial distance, (2) reenters the atmosphere, or (3) fails to reach either condition by a specified number of iterative steps. If the negative test particle path penetrates the specified outer boundary (reaches interplanetary space) the direction of the particle velocity vector at the boundary crossing is specified as asymptotic latitude and longitude (in corresponding geocentric coordinates). If the charged particle re-enters the atmosphere, then the re-entrant coordinates (geocentric latitude and longitude) are given. In this version of the software, an oxygen nuclei (^{16}O) is used as the test particle. Since rigidity is a canonical coordinate, the path of any charged particle having the specified rigidity will be the same.

The software can be adapted to trace the path of any particle of a specified rigidity from a specified position and direction through any model magnetic field as long as the magnetic field is expressed as vectors in the r, θ, ϕ coordinate system.

The 'input' to this program is a data line that specifies the initial position (latitude, longitude and altitude above the earth's surface), direction (azimuth and zenith), and rigidity (momentum per unit charge) along with control parameters to do 'N' trajectory calculations at specified rigidity increments beginning at an initial specified rigidity.

The 'output' of the software is in summary files with one line for each trajectory calculation.

The traditional summary output is called TAPE7 and is in an 80-column "card image" format. This contains the initial conditions (the geodetic latitude, longitude, rigidity, zenith, and azimuth), the final results (asymptotic latitude and longitude), fate of the trajectory, the number of steps in the trajectory calculation, plus a magnetic field identifier.

There is also a second output summary, traditionally called TAPE8, which is in a line printer 132 column format. This contains more detail; the initial conditions (the geodetic and geocentric latitude, the longitude, rigidity, zenith, and azimuth), the final results (asymptotic latitude and longitude, path length (in earth radii), trajectory transit time, time at altitudes under 100 km, number of maximum and minimum in radial distance along the trajectory, the trajectory fate, the number of steps in the calculation, plus a magnetic field identifier.

This structured FORTRAN 77 software assembly consists of a main program and four associated subroutines. The software has extensive internal comments to aid the user in understanding the program. The program and subroutines are:

Program	TJI95T	Main program; primary purpose is control
Subroutine	GDGC	Conversion from geodetic to geocentric coordinates
Subroutine	SINGLTJ	Calculates a particle trajectory
Subroutine	FGRAD	Evaluates the $\mathbf{V} \times \mathbf{B}$ force vectors on the particle
Subroutine	MAGNEW95	Evaluates the vector magnetic field at position r, θ, ϕ

Program Organization:

Each subroutine has a separate unique function. Critical and often used variables are defined in labeled common blocks. The important "working" variables are in the common block WRKVLU. The trigonometric sines and cosines are in common block WRKTSC. Definitions associated with the shape of the ellipsoid representing the surface of the earth are in common block GEOID. We have found that some "super computers" do not allow mixing of real and integer variables in the same common block; therefore there are two additional common blocks associated with subroutine SINGLTJ. These are common block SNGLR (real variables) and common block SNGLI (integer variables)

Accuracy and Precision

It is recommended that the REAL*8 precision always be used. The primary limitation affecting the results is the accuracy of the magnetic field expansion. For reasonably simple trajectories the results should be repeatable, independent of the computer platform used. For long complex trajectories, default compiler options (round off or truncate, and the precision of intrinsic function) begin to affect the result. The calculation procedure includes automatic error checking. The particle acceleration terms are monitored in subroutine SINGLTJ. When significant increases in the force on the particle are noted the step size is reduced and the calculations are continued at smaller step intervals. The quantity BETA ($\beta = v/c$) should be invariant throughout the calculation and is monitored. Changes of BETA exceeding 1 part in 10^5 results in an automatic restart and the trajectory is recalculated at smaller step size increments.

On some "main frames" the intrinsic functions are automatically derived in REAL*8 precision; on some other systems the intrinsic functions are evaluated in a REAL*4 mode unless the double precision argument is specified. In this version intended for the Desktop Computer, all intrinsic functions are specified in the double precision mode; however, we have left a single precision statement "commented out" immediately before each double precision statement. For simple trajectories the user probably cannot note any difference; however, for long complex trajectories, the differences between the use of single precision intrinsic functions and double precision intrinsic functions will become apparent. If a specific rigidity and direction is used for comparison and the position of each trajectory step is monitored, the effect of the small differences between REAL*4 and REAL*8 accumulate and eventually the trajectory path will differ if it is a long complex trajectory.

In the interest of computational speed, the magnetic field calculation routine drops the evaluation of the high order terms when they make an "insignificant" contribution to the total magnetic field. Again for long complex trajectories, these small differences accumulate and the trajectory paths may diverge when different criteria are used for dropping magnetic field expansion.

User Defined Parameters:

Two variables are intended to be user defined. These are FSTEP and LIMIT. Default values have been set in the program, LIMIT = 600,000, and FSTEP = 4×10^8 .

LIMIT is the number of Runge-Kutta steps allowed before a trajectory is declared failed.

FSTEP is the total number of Runge-Kutta steps allowed before the run is terminated.

Simple high rigidity trajectories often require only several hundred steps. Simple trajectories above the upper cutoff rigidity often can be completed in a few thousand steps. Most cosmic ray trajectories will complete in about 10,000 steps. Some quasi-trapped periodic orbits may require more than 100,000 steps. Trapped orbits require an infinite number of steps. Very low rigidity trajectories initiated at high polar latitudes will exhibit the quasi-trapped behavior and probably fail to reach a solution. (The step size criteria is based on the time to travel about one percent of a gyro-distance. Therefore trajectories with many loops require many steps to complete.)

Assuming the user wants to operate in a "batch mode" some job control parameters are needed. This is the quantity FSTEP. Some estimate of the computer speed is necessary. For desktop personal computers this can range from a few hundred steps per second on old obsolete 486 chips to the order of 50,000 steps per second obtainable with current Pentium® III chips operating at approximately 1 GHz clock cycle time. We have found a very significant difference in the program computational speed on the same computer that can be attributed to the efficiency of the object code generated by the compiler. In our testing on desktop platforms we have found that the executable code generated by the COMPAQ® Visual Fortran operates efficiently on a Microsoft® Windows operating system. The worst performing executable code (derived from an old, no longer sold system) ran about five times slower on the same test set of trajectory calculation. It is assumed that workstations will have trajectory computational speeds of the order or at least 10,000 steps per second. The default FSTEP setting will allow a batch run of the order of 10 hours if the program executes at 10,000 Runge-Kutta steps per second.

Program Operation

This program operates in the r, θ, ϕ coordinate system. The variables Y(1), Y(2), and Y(3) are the position vectors in the r, θ, ϕ coordinate system and the variables Y(4), Y(5), and Y(6) are the velocity vectors.

The program initially defines the physical constants used in the calculation and control parameters. In this version it uses system calls to get the date and time for the start of the run. It then enters a control loop beginning with reading a data line to determine the initial position and direction, the specified starting rigidity and how many trajectories to calculate at specified increments.

For each control line read, a call to subroutine GDGC converts the initial geodetic coordinates (map makers coordinates on the earth's surface) to geocentric r, θ, ϕ coordinates. Then the trajectory calculations are done by subroutine SINGLTJ

The control loop continues (read in control line, convert coordinates, do trajectory calculations) until a negative (or zero) value of rigidity is read in. When this occurs, the system calls to get the date and time for the end of the run, and exits.

Labeled Common arguments:

Block name: /WRKVLU/

Arguments in block F(6), Y(6), ERAD, EOMC, VEL, BR, BT, BP

F(6) Array of force and acceleration vectors

Y(6)	Array of position and velocity vectors
ERAD	Average radius of the earth in kilometers
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units)
VEL	Particle velocity in earth radii per second
BP	Value of the $B(\phi)$ magnetic field vector (in units of Gauss)
BR	Value of the $B(r)$ magnetic field vector (in units of Gauss)
BT	Value of the $B(\theta)$ magnetic field vector (in units of Gauss)

Block name: /WRKTSC/
 Arguments in block
 TSY2, TCY2, TSY3, TCY3
 TSY2 Sine of the Y(2) coordinate (theta coordinate)
 TCY2 Cosine of the Y(2) coordinate (theta coordinate)
 TSY3 Sine of the Y(3) coordinate (phi coordinate)
 TCY3 Cosine of the Y(3) coordinate (phi coordinate)

Block name: /TRIG/
 Arguments in block
 PI, RAD, PI02
 PI Value of pi
 RAD Value of degrees in a radian
 PI02 Value of pi/2.0

Block name: /GEOID/
 Arguments in block
 ERADPL, ERECSQ
 ERADPL Polar radius of the earth in kilometers
 ERECSQ Eccentricity of ellipsoid squared

Block name: /SNGLR/
 Arguments in block
 SALT, DISOUT, GCLATD, GDLATD, GLOND, GDAZD, GDZED,
 RY1, RY2, RY3, RHT, TSTEP
 SALT Start altitude of trajectory above surface of geoid
 DISOUT Radial distance (in earth radii) for termination of calculation
 GCLATD Geocentric latitude in degree
 GDLATD Geodetic latitude in degrees
 GLOND East longitude in degrees
 GDAZD Geodetic azimuth in degrees
 GDZED Geodetic zenith in degrees
 RY1 Original start position Y(1), (radial component in the r, θ, ϕ coordinate system)
 RY2 Original start position Y(2), (theta component in the r, θ, ϕ coordinate system)
 RY3 Original start position Y(3), (phi component in the r, θ, ϕ coordinate system)
 RHT Height above geoid where trajectory re-enters the atmosphere
 TSTEP Total number of steps in this run.

Block name: /SNGLI/
 Arguments in block
 LIMIT, NTRAJC, IERRPT
 LIMIT Maximum number of steps before 'failed' trajectory
 NTRAJC Number of trajectories calculated in this run
 IERRPT Integer control for printing diagnostics (normally = 0)

Subroutines called:

GCGC (TCD, TSD)

TCD	Cosine of the rotation angle
TSD	Sine of the rotation angle

SINGLTJ (PC, IRSLT, INDXPC, Y1GC, Y2GC, Y3GC)	
PC	Particle rigidity (in units of GV)
IRSLT	Integer result of trajectory calculation (+1 = Allowed, 0 = Failed, -1 = Re-entrant)
INDXPC	Integer value of PC in units of MV
Y1GC	Y(1) position in geocentric coordinates
Y2GC	Y(2) position in geocentric coordinates
Y3GC	Y(3) position in geocentric coordinates

Dimensioned variables: all in labeled common

F(6)	Array of force and acceleration vectors
Y(6)	Array of position and velocity vectors

Data files: noneOutput files:

TAPE7 from subroutine SINGTJ (80 character summary)
 TAPE8 from subroutine SINGTJ (132 character line printer summary)
 TAPE16 (diagnostic output; if desired, set IERRPT to > 0)

Listing of all variables in program TJI95T

B	Magnitude of the magnetic field (in Gauss) [in COMMON /WRKVLU/]
BP	Value of the $B(\phi)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BR	Value of the $B(r)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BT	Value of the $B(\theta)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
CF	Character variable "F"
DELPC	Increments of rigidity spacing search in control line
DISOUT	Distance (in earth radii) for trajectory termination [in COMMON /SNGLR/]
ERAD	Average radius of the earth in kilometers [in COMMON /WRKVLU/]
ERADPL	Polar radius of the earth in km [in COMMON /GEOID/]
ERECSQ	Eccentricity of ellipsoid squared [in COMMON /GEOID/]
F(6)	Array of "F" values (velocity and acceleration in program coordinates) [in COMMON /WRKVLU/]
GCLATD	Geocentric latitude in degrees (+ = north, south = -) [in COMMON /SNGLR/]
GDAZ	Geodetic azimuth in radians (measured clockwise from north)
GDAZD	Geodetic azimuth in degrees (measured clockwise from north)

GDLATD	Geodetic latitude in degrees (+ = north, south = -) [in COMMON /SNGLR/]
GDZE	Geodetic zenith in radians (0 = vertical)
GDZED	Geodetic zenith in degrees (0 = vertical)
GLOND	Geodetic East longitude in degrees (from Greenwich meridian) [in COMMON /SNGLR/]
IDE LPC	Integer value of rigidity change increment in MV (attempt to avoid round off)
IERRPT	Integer control for printing diagnostics (normally set to 0)
INDEX	Arbitrary index number of input control line (optional)
INDO	Integer control of number of trajectories to calculate
INDXPC	Integer value of rigidity in MV increments (attempt to avoid round off)
IOSTAT	Integer system argument of status of read
IRSLT	Internal result of particle trajectory (+1 = allowed, 0 = failed, -1 = re-entrant)
ISALT	Integer value of start altitude (in km) above geoid surface
LIMIT	Limit of number of steps before trajectory is declared "Failed"
LSTEP	Number of times step size control has been reduced to overcome trajectory error
NDO	Integer control read in (number of trajectories to compute from this control line)
NTRAJC	Number of trajectories in this computer run
PC	Rigidity of particle (in units of GV)
PI	Real value of the quantity Pi (~3.1415926535) [in COMMON /TRIG/]
PIO2	Real value of Pi divided by 2.0 [in COMMON /TRIG/]
RAD	Real value of one radian (~57.29578 degrees) [in COMMON /TRIG/]
RHT	Height above geoid where a trajectory re-entered the atmosphere
RY1	Real value of the starting position of the r coordinate in r, theta, phi coordinates [in COMMON /SNGLR/]]
RY2	Real value of the starting position of the theta coordinate in r, theta, phi coordinates [in COMMON /SNGLR/]]
RY3	Real value of the starting position of the phi coordinate in r, theta, phi coordinates [in COMMON /SNGLR/]]
SALT	Real value of the starting altitude above the surface of the geoid [in COMMON /SNGLR/]]
TCD	Trigonometric cosine of the rotation angle from geodetic to geocentric
TCGDAZ	Trigonometric cosine of the geodetic azimuth (Measured clockwise from north) [in COMMON /SNGLR/]
TCGDZE	Trigonometric cosine of the geodetic zenith (measured clockwise from north)
TCY2	Trigonometric cosine of the vector theta angle in r, theta, phi coordinates [in COMMON /SNGLR/]]
TCY3	Trigonometric cosine of the vector phi angle in r, theta, phi coordinates [in COMMON /SNGLR/]]
TSTEP	Number steps executed in this run
TSY2	Trigonometric sine of the vector theta (theta) angle in r, theta, phi coordinates [in COMMON /SNGLR/]]
TSY3	Value of the trigonometric sine of the vector phi (phi) angle in r, theta, phi coordinates [in COMMON /SNGLR/]]

VEL	Particle velocity in earth radii per second [in COMMON /WRKVLU/]
Y(6)	Array of "Y" values (position and velocity in r, θ, ϕ coordinates) [in COMMON /WRKVLU/]
Y1GC	Starting position r component in geocentric coordinates
Y1GD	Starting position r component in geodetic coordinates
Y2GC	Starting position θ component in geocentric coordinates
Y2GD	Starting position θ component in geodetic coordinates
Y3GC	Starting position ϕ component in geocentric coordinates
Y3GD	Starting position ϕ component in geodetic coordinates

Subroutine GDGC (TCD, TSD)

This subroutine calculates the angle between geodetic and geocentric coordinates. The arguments TCD and TSD are the trigonometric cosine and sine of the rotation angle from a normal from the surface of the geoid (geodetic coordinates) and a radial from the center of the earth (geocentric coordinates). See Appendix B of NSSDC 72-12)

Arguments in call statement

TCD	Cosine of the rotation angle
TSD	Sine of the rotation angle

Labeled Common arguments:

Block name:	/WRKVLU/
Arguments in block	F(6), Y(6), ERAD, EOMC, VEL, BR, BT, BP
F(6)	Array of force and acceleration vectors
Y(6)	Array of position and velocity vectors
ERAD	Average radius of the earth in kilometers
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units)
VEL	Particle velocity in earth radii per second
BP	Value of the $B(\phi)$ magnetic field vector (in units of Gauss)
BR	Value of the $B(r)$ magnetic field vector (in units of Gauss)
BT	Value of the $B(\theta)$ magnetic field vector (in units of Gauss)
Block name:	/WRKTSC/
Arguments in block	TSY2, TCY2, TSY3, TCY3
TSY2	Sine of the $Y(2)$ coordinate (theta coordinate)
TCY2	Cosine of the $Y(2)$ coordinate (theta coordinate)
TSY3	Sine of the $Y(3)$ coordinate (phi coordinate)
TCY3	Cosine of the $Y(3)$ coordinate (phi coordinate)
Block name:	/TRIG/
Arguments in block	PI, RAD, PI02
PI	Value of pi
RAD	Value of degrees in a radian
PI02	Value of $\pi/2.0$
Block name:	/GEOID/
Arguments in block	ERADPL, ERECSQ
ERADPL	Polar radius of the earth in kilometers
ERECSQ	Eccentricity of ellipsoid squared
Block name:	/SNGLR/
Arguments in block	SALT, DISOUT, GCLATD, GDLATD, GLOND, GDAZD, GDZED, RY1, RY2, RY3, RHT, TSTEP
SALT	Start altitude of trajectory above surface of geoid
DISOUT	Radial distance (in earth radii) for termination of calculation
GCLATD	Geocentric latitude in degrees
GDLATD	Geodetic latitude in degrees
GLOND	East longitude in degrees

GDAZD	Geodetic azimuth in degrees
GDZED	Geodetic zenith in degrees
RY1	Original start position Y(1)
RY2	Original start position Y(2)
RY3	Original start position Y(3)
RHT	Height above geoid where trajectory re-enters the atmosphere
TSTEP	Total number of steps in this run.
 Block name:	/SNGLI/
Arguments in block	LIMIT, NTRAJC, IERRPT
LIMIT	Maximum number of steps before 'failed' trajectory
NTRAJC	Number of trajectories calculated in this run
IERRPT	Integer control for printing diagnostics (normally = 0)
 <u>Dimensioned variables:</u>	all in labeled common
F(6)	Array of force and acceleration vectors
Y(6)	Array of position and velocity vectors

Subroutines called: none

Data files: none

Output files: none

Operation:

The shape of the earth used is not a sphere, but an ellipsoid having a specified polar radius, equatorial radius, and eccentricity. When this subroutine is called, it defines the shape of an oblate earth from the polar and equatorial radius, and calculates vectors from a normal on the surface of the ellipsoid to the specified position in geodetic coordinates, at the specified latitude, and determines the vector rotation angle between geodetic coordinates and geocentric coordinates. The sine and cosine of this rotation angle are passed to the calling program. Geodetic latitude is a measure of latitude in a coordinate system normal to the surface of the earth. At a position on or above the surface of the ellipsoid, there is a slight difference between a direction normal to the surface of the ellipsoid and a direction to the geocentric. This difference is latitude dependent. (It is zero at the equator or poles and can be as large as approximately 1/2 of a degree at mid latitudes.) The vector rotation angle allows for direction specification in both geodetic (map) coordinates and geocentric coordinates. This small correction for the direction may be insignificant for some applications, but may be significant for precision calculation in a specific direction at high rigidities.

Data checking: none. The data to describe the shape of the earth are included in the subroutine.

Listing of all variables used in subroutine GDGC of program TJI95T

B	Magnitude of the magnetic field (in Gauss) [in COMMON /WRKVLU/]
BP	Value of the $B(\phi)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]

BR	Value of the B(r) magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BT	Value of the B(θ) magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
DISOUT	Distance (in earth radii) for trajectory termination [in COMMON /SNGLR/]
DISTKM	Starting position geocentric radial distance from geocenter
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units) [in COMMON /WRKVLU/]
ERAD	Average radius of the earth in kilometers [in COMMON /WRKVLU/]
ERADPL	Polar radius of the earth in km [in COMMON /GEOID/]
ERECSQ	Eccentricity of ellipsoid squared [in COMMON /GEOID/]
ERPLSQ	Polar radius of earth (in km) squared
F(6)	Array of "F" values (velocity and acceleration in program coordinates) [in COMMON /WRKVLU/]
GCLATD	Geocentric latitude in degrees (+ = north, south = -) [in COMMON /SNGLR/]
GDAZD	Geodetic azimuth in degrees (measured clockwise from north)
GDCLT	Geodetic co-latitude (in radians)
GDLATD	Geodetic latitude in degrees (+ = north, south = -) [in COMMON /SNGLR/]
GDZED	Geodetic zenith in degrees (0 = vertical)
GLOND	Geodetic East longitude in degrees (from Greenwich meridian) [in COMMON /SNGLR/]
ONE	Intermediate term in computations (see NSSDC ALLMAG description)
PI	Real value of the quantity Pi (~3.1415926535) [in COMMON /TRIG/]
PIO2	Real value of Pi divided by 2.0 [in COMMON /TRIG/]
RAD	Real value of one radian (~57.29578 degrees) [in COMMON /TRIG/]
RHO	Intermediate term in computations (see NSSDC ALLMAG description)
RHT	Height above geoid where a trajectory is declared re-entrant [in COMMON /SNGLR/]
RY1	Real value of the starting position of the r coordinate in r, θ , ϕ coordinates [in COMMON /SNGLR/]
RY2	Real value of the starting position of the theta coordinate in r, θ , ϕ coordinates [in COMMON /SNGLR/]
RY3	Real value of the starting position of the phi coordinate in r, θ , ϕ coordinates [in COMMON /SNGLR/]
SALT	Real value of the starting altitude above the surface of the geoid [in COMMON /SNGLR/]
TCD	Trigonometric cosine of the rotation angle from geodetic to geocentric
TCGDCLT	Trigonometric cosine of the geocentric co-latitude
TCY2	Trigonometric cosine of the vector theta angle in r, θ , ϕ coordinates [in COMMON /SNGLR/]
TCY3	Trigonometric cosine of the vector phi angle in r, θ , ϕ coordinates [in COMMON /SNGLR/]
THREE	Intermediate term in computations (see NSSDC ALLMAG description)

TSD	Trigonometric sine of the rotation angle from geodetic to geocentric
TSGDCLT	Trigonometric sine of the geocentric co-latitude
TSTEP	Number steps executed in this run
TSY2	Trigonometric sine of the vector theta (θ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TSY3	Trigonometric sine of the vector phi (ϕ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TWO	Intermediate term in computations (see NSSDC ALLMAG description)
VEL	Particle velocity in earth radii per second [in COMMON /WRKVLU/]
Y(6)	Array of "Y" values (position and velocity in r, θ, ϕ coordinates) [in COMMON /WRKVLU/]

Reference publication:

ALLMAG, GCAZLMG, LINTRA: Commuter Programs For Geomagnetic Field And Field-Line Calculations, E.G. Stassinopulous and G.D. Mead, NSSDC 72-12, February 1972, NASA, GFSC, Greenbelt MD.

Subroutine SINGLTJ (PC, IRSLT, INDXPC, Y1GC, Y2GC, Y3GC)

This subroutine does the actual trajectory tracing. When called it initially defines control parameters and constants used in the particle tracing and initializes the Runge-Kutta variables to zero. It sets up the initial position and direction, and defines the relativistic parameters relating to the particle total energy and speed.

In this version of the subroutine, an oxygen nuclei (^{16}O) is used as the test particle. By definition a ^{16}O nuclei has a mass of 16 Atomic Mass Units (AMU) and an atomic charge of 8. The mass-energy conversion for one AMU is 0.93114 GeV. If it were desired to modify the program for some other nuclei, such as a proton that has an atomic charge of 1 and atomic mass of 1.0081415 AMU, then the rest mass energy for atomic nuclei must be adjusted.

After the initial definitions, the subroutine then chooses an initial starting step length (a relatively small value) and starts the Runge-Kutta process of tracing the particle trajectory. After each step it goes through an error checking and detection process. If the checks are satisfactory, it determines the particle location with respect to the atmosphere and the outer boundary.

If the charged particle is between the atmosphere and the outer boundary, it adjusts the size of the next step and continues the trajectory tracing until the LIMIT on the number of steps is reached.

If the charged particle is entering the atmosphere, it terminates the calculation.

If the charged particle is less than 100 km above the earth's surface, it maintains a running sum of the time at low altitudes.

If the charged particle is approaching the outer boundary, it adjusts the step size so it penetrates this boundary at small step lengths.

If the charged particle has penetrated the outer boundary at a small step, it computes the final coordinates.

When the particle has reached a solution (allowed or re-entrant) or reached the step limit, it writes out the result and returns to the calling program.

Arguments in call statement

PC, IRSLT, INDXPC, Y1GC, Y2GC, Y3GC

PC	Particle rigidity
IRSLT	Integer result of trajectory (+1 = allowed, 0 = failed, -1 = re-entrant)
INDXPC	Integer value of PC in MV units
Y1GC	Y(1) position in geocentric coordinates
Y2GC	Y(2) position in geocentric coordinates
Y3GC	Y(3) position in geocentric coordinates

Labeled Common arguments:

Block name: /WRKVLU/

Arguments in block F(6), Y(6), ERAD, EOMC, VEL, BR, BT, BP

F(6) Array of force and acceleration vectors

Y(6) Array of position and velocity vectors

ERAD Average radius of the earth in kilometers

EOMC Electronic charge divided by relativistic mass energy equivalent (mixed units)

VEL Particle velocity in earth radii per second

BP Value of the $B(\phi)$ magnetic field vector (in units of Gauss)

BR Value of the $B(r)$ magnetic field vector (in units of Gauss)

BT	Value of the $B(\theta)$ magnetic field vector (in units of Gauss)
Block name:	/WRKTSC/
Arguments in block	TSY2, TCY2, TSY3, TCY3
TSY2	Sine of the Y(2) coordinate (theta coordinate)
TCY2	Cosine of the Y(2) coordinate (theta coordinate)
TSY3	Sine of the Y(3) coordinate (phi coordinate)
TCY3	Cosine of the Y(3) coordinate (phi coordinate)
Block name:	/TRIG/
Arguments in block	PI, RAD, PI02
PI	Value of pi
RAD	Value of degrees in a radian
PI02	Value of pi/2.0
Block name:	/GEOID/
Arguments in block	ERADPL, ERECSQ
ERADPL	polar radius of the earth in kilometers
ERECSQ	Eccentricity of ellipsoid squared
Block name:	/SNGLR/
Arguments in block	SALT, DISOUT, GCLATD, GDLATD, GLOND, GDAZD, GDZED, RY1, RY2, RY3, RHT, TSTEP
SALT	Start altitude of trajectory above surface of geoid
DISOUT	Radial distance (in earth radii) for termination of calculation
GCLATD	Geocentric latitude in degrees
GDLATD	Geodetic latitude in degrees
GLOND	East longitude in degrees
GDAZD	Geodetic azimuth in degrees
GDZED	Geodetic zenith in degrees
RY1	Original start position Y(1), (radial component in the r, θ, φ coordinate system)
RY2	Original start position Y(2), (theta component in the r, θ, φ coordinate system)
RY3	Original start position Y(3), (phi component in the r, θ, φ coordinate system)
RHT	Height above geoid where trajectory re-enters the atmosphere
TSTEP	Total number of steps in this run.
Block name:	/SNGLI/
Arguments in block	LIMIT, NTRAJC, IERRPT
LIMIT	Maximum number of steps before 'failed' trajectory
NTRAJC	Number of trajectories calculated in this run
IERRPT	Integer control for printing diagnostics (normally = 0)
<u>Dimension variables:</u> (not in labeled common)	
P(6), Q(6), R(6), S(6), YB(6), FOLD(6), YOLD(6)	
P(6)	Runge-Kutta variable
Q(6)	Runge-Kutta variable
R(6)	Runge-Kutta variable
S(6)	Runge-Kutta variable
YB(6)	Runge-Kutta variable
FOLD(6)	"F" vectors of previous step
YOLD(6)	"Y" vectors of previous step

Subroutines called: FGRADData files: noneOutput files:

- TAPE7 (80 character summary)
- TAPE8 (132 character line printer summary)
- TAPE16 (diagnostic output; if desired set IERRPT to > 0)

Program Operation:

This program operates in the r, θ, ϕ coordinate system. The variables Y(1), Y(2), and Y(3) are the position vectors in the r, θ, ϕ coordinate system and the variables Y(4), Y(5), and Y(6) are the velocity vectors.

When this subroutine is called, it initially defines control parameters and constants used in the particle path tracing, and initializes the Runge-Kutta variables to zero. It obtains the particle's height with respect to the surface of an oblate earth. It sets up the initial position vectors, Y(1), Y(2) and Y(3), and based on the particle rigidity, sets up velocity vectors, Y(4), Y(5), and Y(6). It then defines the relativistic parameters TENG (total energy), EOMC (charge per relativistic mass/energy equivalent), and GMA (the relativistic parameter of total energy over the rest mass energy). It defines scalar quantities relating to the particle, BETA (the particle speed with respect to light), PVEL (the particle speed in earth radii per second), and HMAX (a maximum step length allowed for this particle rigidity).

Next it defines an initial starting step length (a relatively small value) and starts the Runge-Kutta process of tracing the particle trajectory. Comment cards specifically indicate the Runge-Kutta iteration process, which is the coding between FORTRAN statement numbers 130 and 170. The calls to subroutine FGRAD evaluate the $\mathbf{V} \times \mathbf{B}$ force on the particle during this step. The logic is very similar to that documented in Ralston and Wilf (1960). After each Runge-Kutta iteration step there is an extensive error checking and detection process.

The error checking process begins with a check on the particle speed (BETA), which should remain invariant throughout the trajectory. If the difference between the initial particle speed (BETA) and its current speed (RCKBETA) is greater than EDIF, then the trajectory tracing process is re-initialized (including the NSTEP variable) and the trajectory re-started at a smaller step size selection criteria. Up to five re-starts are allowed before the specific trajectory is declared impossible to calculate, evaluated as "failed", and the path length made negative in order to distinguish it from successful trajectories. In order to attempt to reach a solution the EDIF variable is widened by a factor of two after each successive trajectory failure.

After the error check, then the acceleration of the particle is compared with previous values. We have found that computational errors are most likely to occur when there are rapid changes in the acceleration. If the average change in acceleration exceeds a factor of five, or if any component of the acceleration exceeds a factor of three, then the step length for the next Runge-Kutta step is reduced.

Along the particle path the software checks the particle location with respect to the atmosphere and the outer boundary. If the charged particle is less than 100 km above the earth's surface, it maintains a running sum of the time at low altitudes. If the charged particle is entering the atmosphere, it terminates the calculation.

The next check determines if the particle has penetrated the outer termination boundary. The step length can be relatively large at extreme distances from the earth. If the outer boundary has been penetrated at a large step size, the trajectory is "backed up" and the step size reduced until it penetrates the boundary at a small step size. This results in a more precise determination of the penetration location and can significantly affect the computed asymptotic direction.

If there are no errors and the charged particle is between the atmosphere and the outer boundary, the software adjusts the size of the next step appropriate for the magnitude of the magnetic field (the step size is normally about one percent of the gyro-distance) and continues the trajectory tracing. The basic step length algorithm is:

$$H = ((2.0 * \pi * 33.333 * PC) / (B * \beta * C)) / 100.0$$

where "H" is time in seconds, "PC" is the particle rigidity in GV, "B" is the magnitude of the magnetic field in Gauss and "C" is the speed of light in km/sec.
(A handy formula to remember is the gyro-radius is 33 km per GV per Gauss)

The software initially starts at a trajectory calculation at a small step size and the step size is permitted to grow at a maximum of about 20 percent each step. If the particle trajectory starts to loop to a lower altitude, then the step size is reduced to compensate for the increasing magnitude of the magnetic field. When a loop in the trajectory develops, the acceleration forces increase and step size adjustments are made in the case of a significant increase in the acceleration forces.

During the trajectory tracing, the software notes the number of maximum and minima the trajectory experiences. This information is useful in ascertaining the complexity of the trajectory.

When the particle has reached a solution (allowed or re-entrant) or reached the step limit, the subroutine will write out the result and return to the calling program. The fate is coded in the variable IFATE: (0 = Allowed, 1 = Failed, 2 = Re-entrant, 3 = Failed, but max alt > 6.6 earth radii)

If the trajectory is allowed (penetrates the outer boundary), then the velocity vectors are transformed into asymptotic latitude and longitude. Asymptotic latitude and longitude are the geocentric coordinates the velocity vector would have at infinity. If the trajectory re-enters the atmosphere, then the position coordinates are transformed to geocentric latitude and longitude.

The 'output' of the software is in summary files, on one line for each trajectory calculation.
The output files are:

TAPE7 is in an 80-column "card image" format. This contains the initial conditions (the geodetic latitude, longitude, rigidity, zenith, and azimuth), the final results (asymptotic latitude and longitude, fate and number of steps), and a magnetic field identifier.

TAPE8 is in a line printer 132-column format. This contains more detail; the initial conditions (the geodetic latitude and the geocentric latitude, the longitude, rigidity, zenith, and azimuth), the final results (asymptotic latitude and longitude, path length,

trajectory time, time at altitudes under 100 km, number of maximum and minimum in trajectory, fate, and number of steps), and a magnetic field identifier.

TAPE16 is a diagnostic output, including a record of restarts due to BETA checks or trajectory failures.

See the examples section for samples of the output.

Possible Additions for Trajectory Plotting

If it is desired to plot a trajectory, the position variables Y(1), Y(2), and Y(3) must be stored after each Runge-Kutta step in a suitable array. The task of adding such a modification should be straightforward and is left to the individual program user.

Listing of all variables in subroutine SINGLTJ

ACCER	Magnitude of current value of the particle acceleration
ACCOLD	Magnitude of last value of the particle acceleration
AFOLD	Absolute value of FOLD
AHLT	Variable to control step size at high latitude
ANUC	Atomic number of number of nucleons in atom
ATRG1	Intermediate value to compute asymptotic latitude (avoid 0/0 problem)
ATRG2	Intermediate value to compute asymptotic latitude (avoid 0/0 problem)
AZD	Azimuth angle in degrees (measured clockwise from north)
B	Magnitude of the magnetic field (in Gauss) [in COMMON /WRKVLU/]
BETA	Particle speed as fraction of light speed ($\beta = v/c$ where c is speed of light)
BETAST	Control variable for reducing step size if error has occurred
BP	Value of the $B(\phi)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BR	Value of the $B(r)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BT	Value of the $B(\theta)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
CF	Character variable "F"
CNAME	Character variable (up to 6 characters) identifying magnetic field used
CR	Character variable "R"
DELACC	Change in particle acceleration from previous step
DISCK	Step length control variable to approach boundary at 10 % increments
DISOUT	Distance (in earth radii) for trajectory termination [in COMMON /SNGLR/]
DISTR	Radial distance (in earth radii) from current distance to termination boundary
EDIF	Variation in β allowed before error declared
EMCSQ	Mass energy equivalent for a AMU (0.931141 GeV)
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units) [in COMMON /WRKVLU/]
ERAD	Average radius of the earth in kilometers [in COMMON /WRKVLU/]
ERADPL	Polar radius of the earth in km [in COMMON /GEOID/]
ERECSQ	Eccentricity of ellipsoid squared [in COMMON /GEOID/]
F(6)	Array of "F" values (velocity and acceleration in program coordinates)

		[in COMMON /WRKVLU/]
FOLD(6)		Array of "F" values (velocity and acceleration) from previous step
FASLAT		Asymptotic latitude (in degrees)
FASLON		Asymptotic longitude (in degrees east of the Greenwich meridian)
GCLATD		Geocentric latitude in degrees (+ = north, south = -) [in COMMON /SNGLR/]
GDAZD		Geodetic azimuth in degrees (measured clockwise from north)
GDLATD		Geodetic latitude in degrees (+ = north, south = -) [in COMMON /SNGLR/]
GDZED		Geodetic zenith in degrees (0 = vertical)
GLOND		Geodetic East longitude in degrees (from Greenwich meridian)
GMA		[in COMMON /SNGLR/]
GRNDKM		Relativistic factor (total energy/rest energy)
		Altitude above surface of earth at this latitude (in km)
H		Runge-Kutta step size (in seconds)
HB		Preliminary value of step size for $\beta = 1$
HCK		Control to limit step size growth to 20%
HCNG		Change of step size from previous step
HMAX		Maximum value of step size allowed
HOLD		Value of previous Runge-Kutta step size
HSNEK		Control to approach 90% of distance to boundary
HSTART		Starting step size value (deliberately made small)
I		Index variable in do loops
IAZ		Integer value of azimuth (measured counter clockwise from north)
ICK		Index for checking acceleration growth
IERRPT		Integer control for printing diagnostics (normally set to 0)
IFATE		Integer fate of particle trajectory (0 = Allowed, 1 = Failed, 2 = Re-entrant 3 = Failed, but max alt > 6.6 earth radii)
INDXPC		Index of particle rigidity in MV
IRT		Integer control for writing results (+1 = Allowed, 0 = Failed, -1 = Re-entrant)
IRSLT		Internal result of particle trajectory (+1 = Allowed, 0 = Failed, -1 = Re-entrant)
ISALT		Integer value of start altitude (in km) above earth surface
IZE		Integer value of zenith angle (in degrees)
KBF		Number of failed attempts to trace this trajectory
LIMIT		Limit of number of steps before trajectory declared "Failed"
LSTEP		Number of times the step size control reduced to overcome trajectory error
NMAX		Number of maxima in complex trajectory path
NMIN		Number of minima in complex trajectory path
NSTEP		Number of steps in current trajectory
NSTEPT		Temporary variable that can be used to print out first 1000 steps
NTRAJC		Number of trajectories in this computer run
P(6)		Array of intermediate values used in Runge-Kutta integration
PATH		Total distance of trajectory path from start to termination
PC		Rigidity of particle (in units of GV)
PI		Real value of the quantity Pi (~3.1415926535) [in COMMON /TRIG/]
PIO2		Real value of Pi divided by 2.0 [in COMMON /TRIG/]

PSALT	Current particle distance from ground (used for re-entrant calculations)
PTCY2	Absolute value of cosine Y(2) (used in control of polar step size)
PVEL	Particle velocity (in earth radii per second)
R(6)	Array of intermediate values used in Runge-Kutta integration
R100KM	Y(1) distance of 100 km altitude at this latitude
R120KM	Y(1) distance of 120 km altitude at this latitude
RAD	Real value of one radian (~57.29578 degrees) [in COMMON /TRIG/]
RC1O6	Constant in Runge-Kutta integration (1.0/6.0)
RCKBETA	Current value of particle β after this step
RENLAT	Latitude of re-entrant particle intersection with atmosphere
RENLON	Longitude of re-entrant particle intersection with atmosphere
RFA	Ratio of acceleration magnitude between current step and last step
RFCK	Ratio of acceleration component between current step and last step
RHT	Height above geoid where a trajectory re-entered the atmosphere
RY1	Real value of the starting position of the r coordinate in r, θ, ϕ coordinates [in COMMON /SNGLR/]
RY2	Real value of the starting position of the theta coordinate in r, θ, ϕ coordinates [in COMMON /SNGLR/]
RY3	Real value of the starting position of the phi coordinate in r, θ, ϕ coordinates [in COMMON /SNGLR/]
S(6)	Array of intermediate values used in Runge-Kutta integration
SALT	Real value of the starting altitude above the surface of the geoid [in COMMON /SNGLR/]
SR2	Runge-Kutta constant (square root of 2.0)
TAU	Time (in seconds) for a trajectory transit from start to termination
TBETA	Difference between current value of β and starting value of β
TCY2	Trigonometric cosine of the vector theta (θ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TCY3	Trigonometric cosine of the vector phi (ϕ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TENG	Total energy of particle (kinetic energy plus rest mass energy)
TMS2O2	Runge-Kutta constant (2.0 - SR2/2.0)
TPS2O2	Runge-Kutta constant ((2.0 + SR2/2.0))
TSTEP	Number steps executed in this run
TSY2	Trigonometric sine of the vector theta (θ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TSY2SQ	Square of TSY2
TSY3	Value of the trigonometric sine of the vector phi (ϕ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TU100	Time (in seconds) the particle is under 100 km altitude
VEL	Particle velocity in earth radii per second [in COMMON /WRKVLU/]
Y(6)	Array of "Y" values (position and velocity in r, θ, ϕ coordinates) [in COMMON /WRKVLU/]
YB(6)	Array of intermediate values used in Runge-Kutta integration
Y10	Y(1) radial coordinate for re-entrant distance at this latitude

Y1GC	Starting position r component in geocentric coordinates
Y2GC	Starting position θ component in geocentric coordinates
Y3GC	Starting position ϕ component in geocentric coordinates
YDA5	Intermediate value for computing asymptotic latitude
YMAX	Maximum radial distance attained by trajectory
ZCHARGE	Atomic charge number
ZED	Zenith angle in degrees

Reference publication:

Ralston, A, and Wilf, S.H., Mathematical Models for Digital Computers, John Wiley and Sons, New York, 1960.

Subroutine FGRAD

This subroutine calculates the $\mathbf{V} \times \mathbf{B}$ force on the charged particle.

Arguments in call statement: none. All in labeled common

Labeled Common arguments:

Block name:	/WRKVLU/
Arguments in block	F(6), Y(6), ERAD, EOMC, VEL, BR, BT, BP
F(6)	Array of force and acceleration vectors
Y(6)	Array of position and velocity vectors
ERAD	Average radius of the earth in kilometers
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units)
VEL	Particle velocity in earth radii per second
BP	Value of the $B(\phi)$ magnetic field vector (in units of Gauss)
BR	Value of the $B(r)$ magnetic field vector (in units of Gauss)
BT	Value of the $B(\theta)$ magnetic field vector (in units of Gauss)
Block name:	/WRKTSC/
Arguments in block	TSY2, TCY2, TSY3, TCY3
TSY2	Sine of the $Y(2)$ coordinate (theta coordinate)
TCY2	Cosine of the $Y(2)$ coordinate (theta coordinate)
TSY3	Sine of the $Y(3)$ coordinate (phi coordinate)
TCY3	Cosine of the $Y(3)$ coordinate (phi coordinate)

Subroutines called:

MAGNEW95

(All arguments are in labeled common /WRKVLU/ and /WRKTSC/)

Data files: none

Output files: none

Program Operation:

When this subroutine is called, the force vectors, ((F(1), F(2), F(3)) are defined. The sine and cosine of the $Y(2)$ coordinates are determined. The magnetic field vectors (BR, BT, BP) at the particle position (Y(1), Y(2), Y(3)) are obtained by the call to subroutine MAGNEW95. Then the acceleration vectors (F(4), F(5), F(6)) are calculated.

Listing of all variables in subroutine FGRAD in program TJI95T

B Magnitude of the magnetic field (in Gauss) [in COMMON /WRKVLU/]

BP Value of the $B(\phi)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]

BR	Value of the B(r) magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BT	Value of the B(θ) magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units) [in COMMON /WRKVLU/]
ERAD	Average radius of the earth in kilometers [in COMMON /WRKVLU/]
F(6)	Array of "F" values (velocity and acceleration in program coordinates) [in COMMON /WRKVLU/]
SQY6	Intermediate term $(Y_6)^* Y_6 / Y_1$
TAY2	Intermediate term $T\bar{S}Y_2 / T\bar{S}C_2$
TCY2	Trigonometric cosine of the vector theta angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TCY3	Trigonometric cosine of the vector phi angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TSY2	Trigonometric sine of the vector theta (θ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TSY3	Trigonometric sine of the vector phi (ϕ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
VEL	Particle velocity in earth radii per second [in COMMON /WRKVLU/]
Y(6)	Array of "Y" values (position and velocity in r, θ, ϕ coordinates) [in COMMON /WRKVLU/]
Y5OY1	Intermediate term Y_5 / Y_1

Subroutine MAGNEW95

This is the magnetic field evaluation program containing the IGRF95 magnetic field model. The normalized magnetic field coefficients have been pre-processed and loaded in as data statements. This is a serial computation designed to repeatedly evaluate the same magnetic field model at different positions in space at optimum efficiency. Each term of the magnetic field expansion is the derivative of the previous term.

Arguments in call statement: none. (All in labeled common)

Labeled Common arguments:

Block name:	/WRKVLU/
Arguments in block	F(6), Y(6), ERAD, EOMC, VEL, BR, BT, BP
F(6)	Array of force and acceleration vectors
Y(6)	Array of position and velocity vectors
ERAD	Average radius of the earth in kilometers
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units)
VEL	Particle velocity in earth radii per second
BP	Value of the $B(\phi)$ magnetic field vector (in units of Gauss)
BR	Value of the $B(r)$ magnetic field vector (in units of Gauss)
BT	Value of the $B(\theta)$ magnetic field vector (in units of Gauss)
Block name:	/WRKTSC/
Arguments in block	TSY2, TCY2, TSY3, TCY3
TSY2	Sine of the $Y(2)$ coordinate (theta coordinate)
TCY2	Cosine of the $Y(2)$ coordinate (theta coordinate)
TSY3	Sine of the $Y(3)$ coordinate (phi coordinate)
TCY3	Cosine of the $Y(3)$ coordinate (phi coordinate)

Dimensioned variables: (not in labeled common)

G(11,11)	Normalized coefficients ordered for fast serial computation
BM(11)	Values to determine if the expansion should terminated at order N

Program Operation:

This subroutine is designed for efficient serial computation of the earth's main magnetic field. This procedure expands the terms into FORTRAN coding (resulting in pages and pages of FORTRAN code) and then evaluates the normalized field coefficients. The result of this serial expansion is approximately an order of magnitude speed increase over the recursion method which is much more compact in program size but requires the expansion of the Legender polynomials each time the subroutine is called.

When this subroutine is called it checks the value of the variable JDATA. If JDATA is not 77, then it loads in the data coefficients; otherwise it proceeds directly to the magnetic field evaluation. The array G(11,11) contains the pre-processed and normalized magnetic field model coefficients ordered for fast serial computation. The array BM(11) contains a set of values that determine if the expansion should be

terminated above a specified order because the additional contribution of the magnetic field would not be significant. (This is an additional technique to speed up the computation.)

At the beginning of the magnetic field determination, the sine and cosine of Y(3) (the phi coordinate) are calculated. (This has not been done since the completion of the last Runge-Kutta step.) It then determines the value of the AR variable; AR is reciprocal of the radial distance, (Y(1)) ,in earth radii. Each order of the field expansion requires an evaluation of AR^N where N is the order of the field expansion. (The N = 1 term, a description of a monopole magnetic field, is zero.) The first magnetic field evaluation designated by N = 2 is the dipole field component. Each subsequent order of expansion evaluates the contribution of the next order and adds this to the contribution of the previous orders. The final computation converts the magnetic field vectors to units of Gauss. (One Gauss - 10⁵ Nt). A more detailed description of the process is given in NSSDC DATA USERS NOTE 68-11.

Listing of all variables in subroutine MAGNEW95

AOR	Temporary value of radial distance (in earth radii) to the N th power
AR	1.0/Radial distance (in earth radii) [AR = 1.0/Y(1)]
B	Magnitude of the magnetic field (in Gauss) [in COMMON /WRKVLU/]
BERR	Variable for dropping terms if field magnitude is less than significant value (set to 0.001 for improved accuracy)
BM(11)	Array of check values for dropping terms of magnetic field expansion
BP	Value of the B(ϕ) magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BR	Value of the B(r) magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BT	Value of the B(θ) magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
DP101	Derivative of the polynomial (10,1) term
DP102	Derivative of the polynomial (10,2) term
DP103	Derivative of the polynomial (10,3) term
DP104	Derivative of the polynomial (10,4) term
DP105	Derivative of the polynomial (10,5) term
DP106	Derivative of the polynomial (10,6) term
DP107	Derivative of the polynomial (10,7) term
DP108	Derivative of the polynomial (10,8) term
DP109	Derivative of the polynomial (10,9) term
DP111	Derivative of the polynomial (11,1) term
DP110	Derivative of the polynomial (11,10) term
DP1111	Derivative of the polynomial (11,11) term
DP112	Derivative of the polynomial (11,2) term
DP113	Derivative of the polynomial (11,3) term
DP114	Derivative of the polynomial (11,4) term
DP115	Derivative of the polynomial (11,5) term
DP116	Derivative of the polynomial (11,6) term
DP117	Derivative of the polynomial (11,7) term
DP118	Derivative of the polynomial (11,8) term
DP119	Derivative of the polynomial (11,9) term
DP21	Derivative of the polynomial (2,1) term
DP22	Derivative of the polynomial (2,2) term
DP31	Derivative of the polynomial (3,1) term
DP32	Derivative of the polynomial (3,2) term
DP33	Derivative of the polynomial (3,3) term

DP41	Derivative of the polynomial (4,1) term
DP42	Derivative of the polynomial (4,2) term
DP43	Derivative of the polynomial (4,3) term
DP44	Derivative of the polynomial (4,4) term
DP51	Derivative of the polynomial (5,1) term
DP52	Derivative of the polynomial (5,2) term
DP53	Derivative of the polynomial (5,3) term
DP54	Derivative of the polynomial (5,4) term
DP55	Derivative of the polynomial (5,5) term
DP61	Derivative of the polynomial (6,1) term
DP62	Derivative of the polynomial (6,2) term
DP63	Derivative of the polynomial (6,3) term
DP64	Derivative of the polynomial (6,4) term
DP65	Derivative of the polynomial (6,5) term
DP66	Derivative of the polynomial (6,6) term
DP71	Derivative of the polynomial (7,1) term
DP72	Derivative of the polynomial (7,2) term
DP73	Derivative of the polynomial (7,3) term
DP74	Derivative of the polynomial (7,4) term
DP75	Derivative of the polynomial (7,5) term
DP76	Derivative of the polynomial (7,6) term
DP77	Derivative of the polynomial (7,7) term
DP81	Derivative of the polynomial (8,1) term
DP82	Derivative of the polynomial (8,2) term
DP83	Derivative of the polynomial (8,3) term
DP84	Derivative of the polynomial (8,4) term
DP85	Derivative of the polynomial (8,5) term
DP86	Derivative of the polynomial (8,6) term
DP87	Derivative of the polynomial (8,7) term
DP88	Derivative of the polynomial (8,8) term
DP91	Derivative of the polynomial (9,1) term
DP92	Derivative of the polynomial (9,2) term
DP93	Derivative of the polynomial (9,3) term
DP94	Derivative of the polynomial (9,4) term
DP95	Derivative of the polynomial (9,5) term
DP96	Derivative of the polynomial (9,6) term
DP97	Derivative of the polynomial (9,7) term
DP98	Derivative of the polynomial (9,8) term
DP99	Derivative of the polynomial (9,9) term
DP1010	Derivative of the polynomial (10,10) term
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units) (not used in this subroutine) [in COMMON /WRKVLU/]
ERAD	Average radius of the earth in kilometers (not used in this subroutine) [in COMMON /WRKVLU/]
ERR	Intermediate term used to evaluate when higher order terms can be dropped
F(6)	Array of "F" values (velocity and acceleration in program coordinates) [in COMMON /WRKVLU/]
G(11,11)	Array of normalized magnetic field coefficients ordered for fast computation

GMSUM	Check value to assure that proper magnetic field coefficients are loaded
GSUM	Check data for testing that proper magnetic field coefficients are loaded
JDATA	Integer test value for loading in data statement for magnetic field coefficients
JMAG	Integer order of magnetic field expansion (N+1)
L	Intermediate index for data checking
M	Intermediate index for data checking
MGNMAX	Integer value of maximum order of magnetic field expansion
P101	Polynomial (10,1) term
P1010	Polynomial (10,10) term
P102	Polynomial (10,2) term
P103	Polynomial (10,3) term
P104	Polynomial (10,4) term
P105	Polynomial (10,5) term
P106	Polynomial (10,6) term
P107	Polynomial (10,7) term
P108	Polynomial (10,8) term
P109	Polynomial (10,9) term
P111	Polynomial (11,1) term
P1110	Polynomial (11,10) term
P1111	Polynomial (11,11) term
P112	Polynomial (11,2) term
P113	Polynomial (11,3) term
P114	Polynomial (11,4) term
P115	Polynomial (11,5) term
P116	Polynomial (11,6) term
P117	Polynomial (11,7) term
P118	Polynomial (11,8) term
P119	Polynomial (11,9) term
P21	Polynomial (2,1) term
P22	Polynomial (2,2) term
P31	Polynomial (3,1) term
P32	Polynomial (3,2) term
P33	Polynomial (3,3) term
P41	Polynomial (4,1) term
P42	Polynomial (4,2) term
P43	Polynomial (4,3) term
P44	Polynomial (4,4) term
P51	Polynomial (5,1) term
P52	Polynomial (5,2) term
P53	Polynomial (5,3) term
P54	Polynomial (5,4) term
P55	Polynomial (5,5) term
P61	Polynomial (6,1) term
P62	Polynomial (6,2) term
P63	Polynomial (6,3) term
P64	Polynomial (6,4) term
P65	Polynomial (6,5) term

P66	Polynomial (6,6) term
P71	Polynomial (7,1) term
P72	Polynomial (7,2) term
P73	Polynomial (7,3) term
P74	Polynomial (7,4) term
P75	Polynomial (7,5) term
P76	Polynomial (7,6) term
P77	Polynomial (7,7) term
P81	Polynomial (8,1) term
P82	Polynomial (8,2) term
P83	Polynomial (8,3) term
P84	Polynomial (8,4) term
P85	Polynomial (8,5) term
P86	Polynomial (8,6) term
P87	Polynomial (8,7) term
P88	Polynomial (8,8) term
P91	Polynomial (9,1) term
P92	Polynomial (9,2) term
P93	Polynomial (9,3) term
P94	Polynomial (9,4) term
P95	Polynomial (9,5) term
P96	Polynomial (9,6) term
P97	Polynomial (9,7) term
P98	Polynomial (9,8) term
P99	Polynomial (9,9) term
RC10	Intermediate value (N=10) in magnetic field expansion
RC11	Intermediate value (N=11) in magnetic field expansion
RC2	Intermediate value (N=2) in magnetic field expansion
RC3	Intermediate value (N=3) in magnetic field expansion
RC4	Intermediate value (N=4) in magnetic field expansion
RC5	Intermediate value (N=5) in magnetic field expansion
RC6	Intermediate value (N=6) in magnetic field expansion
RC7	Intermediate value (N=7) in magnetic field expansion
RC8	Intermediate value (N=8) in magnetic field expansion
RC9	Intermediate value (N=9) in magnetic field expansion
TCP10	Trigonometric cosine function for the P10 term
TCP11	Trigonometric cosine function for the P11 term
TCP2	Trigonometric cosine function for the P2 term
TCP3	Trigonometric cosine function for the P3 term
TCP4	Trigonometric cosine function for the P4 term
TCP5	Trigonometric cosine function for the P5 term
TCP6	Trigonometric cosine function for the P6 term
TCP7	Trigonometric cosine function for the P7 term
TCP8	Trigonometric cosine function for the P8 term
TCP9	Trigonometric cosine function for the P9 term
TCY2	Trigonometric cosine of the vector theta angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TCY3	Trigonometric cosine of the vector phi angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]

TSP10	Trigonometric sine function for the P10 term
TSP11	Trigonometric sine function for the P11 term
TSP2	Trigonometric sine function for the P2 term
TSP3	Trigonometric sine function for the P3 term
TSP4	Trigonometric sine function for the P4 term
TSP5	Trigonometric sine function for the P5 term
TSP6	Trigonometric sine function for the P6 term
TSP7	Trigonometric sine function for the P7 term
TSP8	Trigonometric sine function for the P8 term
TSP9	Trigonometric sine function for the P9 term
TSY2	Trigonometric sine of the vector theta angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TSY3	Trigonometric sine of the vector phi angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
VEL	Particle velocity in earth radii per second (not used in this subroutine) [in COMMON /WRKVLU/]
Y(6)	Array of "Y" values (position and velocity in r, θ, ϕ coordinates) [in COMMON /WRKVLU/]

Reference publication:

Computation of the Main Geomagnetic Field for Spherical Harmonic Expansion, Data Users Note
NSSDC 68-11, February 1968, NSDC, NASA, GFSC, Greenbelt MD.

DESCRIPTION OF PROGRAM TJALLMAG

This software package is self-contained and capable of being compiled and executed on a variety of platforms ranging from a personal computer to large scale "super computers". The software is written in FORTRAN 77. The software is designed to efficiently compute the trajectory of an energetic charged particle of a specified momentum per unit charge (rigidity) through a model magnetic field. For cosmic ray access to the earth, the geocenter becomes the origin of the coordinate system. All calculations are done in the r, θ, ϕ coordinate system (a right-handed, orthogonal coordinate system). The magnetic field subroutine included in this software is the NASA NSSDC ALLMAG program which has the option of including a number of magnetic field models. This version has the control (MODEL = 14) which specifies the IGRF95 model of the earth's magnetic field.

In its usual mode of computing the path of cosmic ray trajectories in a model of the earth's magnetic field, we utilize this program to determine the path of a cosmic ray (a positively charged particle) from interplanetary space arriving at the earth at a specified position and direction. To accomplish this, a negatively charged particle is 'launched' from the 'top' of the atmosphere at a specified position (latitude and longitude) in a specific direction (zenith and azimuth), and its path traced through the model magnetic field until it either (1) reaches a specified radial distance, (2) reenters the atmosphere, or (3) fails to reach either condition by a specified number of iterative steps. If the negative test particle path penetrates the specified outer boundary (reaches interplanetary space) the direction of the particle velocity vector at the boundary crossing is specified as asymptotic latitude and longitude (in corresponding geocentric coordinates). If the charged particle re-enters the atmosphere, then the re-entrant coordinates (geocentric latitude and longitude) are given. In this version of the software, an oxygen nuclei (^{16}O) is used as the test particle. Since rigidity is a canonical coordinate, the path of any charged particle having the specified rigidity will be the same.

The software can be adapted to trace the path of any particle of a specified rigidity from a specified position and direction through any model magnetic field as long as the magnetic field is expressed as vectors in the r, θ, ϕ coordinate system.

The 'input' to this program is a data line that specifies the initial position (latitude, longitude and altitude above the earth's surface), direction (azimuth and zenith), and rigidity (momentum per unit charge) along with control parameters to do 'N' trajectory calculations at specified rigidity increments beginning at an initial specified rigidity.

The 'output' of the software is in summary files with one line for each trajectory calculation. The traditional summary output is called TAPE7 and is in an 80-column "card image" format. This contains the initial conditions (the geodetic latitude, longitude, rigidity, zenith, and azimuth), the final results (asymptotic latitude and longitude), fate of the trajectory, the number of steps in the trajectory calculation, plus a magnetic field identifier.

There is also a second output summary, traditionally called TAPE8, which is in a line printer 132 column format. This contains more detail; the initial conditions (the geodetic and geocentric latitude, the longitude, rigidity, zenith, and azimuth), the final results (asymptotic latitude and longitude, path length (in earth radii), trajectory transit time, time at altitudes under 100 km, number of maximum and minimum in radial distance along the trajectory, the trajectory fate, the number of steps in the calculation, plus a magnetic field identifier.

This structured FORTRAN 77 software assembly consists of a main program and four associated subroutines. The software has extensive internal comments to aid the user in understanding the program. The program and subroutines are:

Program	TJALLMAG	Main program; primary purpose is control
Subroutine	GDGC	Conversion from geodetic to geocentric coordinates
Subroutine	SINGLTJ	Calculates a particle trajectory
Subroutine	FGRADA	Evaluates the $V \times B$ force vectors on the particle
Subroutine	ALLMAG	Calculates the vector magnetic field at position r, θ, ϕ

Program Organization:

Each subroutine has a separate unique function. Critical and often used variables are defined in labeled common blocks. The important "working" variables are in the common block WRKVLU. The trigonometric sines and cosines are in common block WRKTSC. Definitions associated with the shape of the ellipsoid representing the surface of the earth are in common block GEOID. We have found that some "super computers" do not allow mixing of real and integer variables in the same common block; therefore there are two additional common blocks associated with subroutine SINGLTJ. These are common block SNGLR (real variables) and common block SNGLI (integer variables)

Accuracy and Precision

It is recommended that the REAL*8 precision always be used. The primary limitation affecting the results is the accuracy of the magnetic field expansion. For reasonably simple trajectories the results should be repeatable, independent of the computer platform used. For long complex trajectories, default compiler options (round off or truncate, and the precision of intrinsic function) begin to affect the result. The calculation procedure includes automatic error checking. The particle acceleration terms are monitored in subroutine SINGLTJ. When significant increases in the force on the particle are noted the step size is reduced and the calculations are continued at smaller step intervals. The quantity BETA ($\beta = v/c$) should be invariant throughout the calculation and is monitored. Changes of BETA exceeding 1 part in 10^5 results in an automatic restart and the trajectory is recalculated at smaller step size increments.

On some "main frames" the intrinsic functions are automatically derived in REAL*8 precision; on some other systems the intrinsic functions are evaluated in a REAL*4 mode unless the double precision argument is specified. In this version intended for the Desktop Computer, all intrinsic functions are specified in the double precision mode; however, we have left a single precision statement "commented out" immediately before each double precision statement. For simple trajectories the user probably cannot note any difference; however, for long complex trajectories, the differences between the use of single precision intrinsic functions and double precision intrinsic functions will become apparent. If a specific rigidity and direction is used for comparison and the position of each trajectory step is monitored, the effect of the small differences between REAL*4 and REAL*8 accumulate and eventually the trajectory path will differ if it is a long complex trajectory.

User Defined Parameters:

Two variables are intended to be user defined. These are FSTEP and LIMIT. Default values have been set in the program, LIMIT = 600,000, and FSTEP = 4×10^8 .
LIMIT is the number of Runge-Kutta steps allowed before a trajectory is declared failed.

FSTEP is the total number of Runge-Kutta steps allowed before the run is terminated. Simple high rigidity trajectories often require only several hundred steps. Simple trajectories above the upper cutoff rigidity often can be completed in a few thousand steps. Most cosmic ray trajectories will complete in about 10,000 steps. Some quasi-trapped periodic orbits may require more than 100,000 steps. Trapped orbits require an infinite number of steps. Very low rigidity trajectories initiated at high polar latitudes will exhibit the quasi-trapped behavior and probably fail to reach a solution. (The step size criteria is based on the time to travel about one percent of a gyro-distance. Therefore trajectories with many loops require many steps to complete.)

Assuming the user wants to operate in a "batch mode" some job control parameters are needed. This is the quantity FSTEP. Some estimate of the computer speed is necessary. For desktop personal computers this can range from a few hundred steps per second on old obsolete 486 chips to the order of 50,000 steps per second obtainable with current Pentium® III chips operating at approximately 1 GHz clock cycle time. We have found a very significant difference in the program computational speed on the same computer that can be attributed to the efficiency of the object code generated by the compiler. In our testing on desktop platforms we have found that the executable code generated by the COMPAQ® Visual Fortran operates efficiently on a Microsoft® Windows operating system. The worst performing executable code (derived from an old, no longer sold system) ran about five times slower on the same test set of trajectory calculation. It is assumed that workstations will have trajectory computational speeds of the order or at least 10,000 steps per second. The default FSTEP setting will allow a batch run of the order of 10 hours if the program executes at 10,000 Runge-Kutta steps per second.

Program Operation

This program operates in the r, θ, ϕ coordinate system. The variables Y(1), Y(2), and Y(3) are the position vectors in the r, θ, ϕ coordinate system and the variables Y(4), Y(5), and Y(6) are the velocity vectors.

The program initially defines the physical constants used in the calculation and control parameters. It then enters a control loop beginning with reading a data line to determine the initial position and direction, the specified starting rigidity and how many trajectories to calculate at specified increments.

For each control line read, a call to subroutine GDGC converts the initial geodetic coordinates (map makers coordinates on the earth's surface) to geocentric r, θ, ϕ coordinates. Then the trajectory calculations are done by subroutine SINGLTJ

The control loop continues (read in control line, convert coordinates, do trajectory calculations) until a negative (or zero) value of rigidity is read in. When this occurs the program terminates.

Labeled Common arguments:

Block name:	/WRKVLU/
Arguments in block	F(6), Y(6), ERAD, EOMC, VEL, BR, BT, BP
F(6)	Array of force and acceleration vectors
Y(6)	Array of position and velocity vectors
ERAD	Average radius of the earth in kilometers
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units)
VEL	Particle velocity in earth radii per second
BP	Value of the $B(\phi)$ magnetic field vector (in units of Gauss)

BR	Value of the $B(r)$ magnetic field vector (in units of Gauss)
BT	Value of the $B(\theta)$ magnetic field vector (in units of Gauss)
Block name:	/WRKTSC/
Arguments in block	TSY2, TCY2, TSY3, TCY3
TSY2	Sine of the Y(2) coordinate (theta coordinate)
TCY2	Cosine of the Y(2) coordinate (theta coordinate)
TSY3	Sine of the Y(3) coordinate (phi coordinate)
TCY3	Cosine of the Y(3) coordinate (phi coordinate)
Block name:	/TRIG/
Arguments in block	PI, RAD, PI02
PI	Value of pi
RAD	Value of degrees in a radian
PI02	Value of $\pi/2.0$
Block name:	/GEOID/
Arguments in block	ERADPL, ERECSQ
ERADPL	Polar radius of the earth in kilometers
ERECSQ	Eccentricity of ellipsoid squared
Block name:	/SNGLR/
Arguments in block	SALT, DISOUT, GCLATD, GDLATD, GLOND, GDAZD, GDZED, RY1, RY2, RY3, RHT, TSTEP
SALT	Start altitude of trajectory above surface of geoid
DISOUT	Radial distance (in earth radii) for termination of calculation
GCLATD	Geocentric latitude in degree
GDLATD	Geodetic latitude in degrees
GLOND	East longitude in degrees
GDAZD	Geodetic azimuth in degrees
GDZED	Geodetic zenith in degrees
RY1	Original start position Y(1), (radial component in the r, θ, ϕ coordinate system)
RY2	Original start position Y(2), (theta component in the r, θ, ϕ coordinate system)
RY3	Original start position Y(3), (phi component in the r, θ, ϕ coordinate system)
RHT	Height above geoid where trajectory re-enters the atmosphere
TSTEP	Total number of steps in this run.
Block name:	/SNGLI/
Arguments in block	LIMIT, NTRAJC, IERRPT
LIMIT	Maximum number of steps before 'failed' trajectory
NTRAJC	Number of trajectories calculated in this run
IERRPT	Integer control for printing diagnostics (normally = 0)

Subroutines called:

GCGC (TCD, TSD)	
TCD	Cosine of the rotation angle
TSD	Sine of the rotation angle

SINGLTJ (PC, IRSLT, INDXPC, Y1GC, Y2GC, Y3GC)

PC	Particle rigidity (in units of GV)
IRSLT	Integer result of trajectory calculation (+1 = Allowed, 0 = Failed, -1 = Re-entrant)
INDXPC	Integer value of PC in units of MV
Y1GC	Y(1) position in geocentric coordinates
Y2GC	Y(2) position in geocentric coordinates
Y3GC	Y(3) position in geocentric coordinates

Dimensioned variables: all in labeled common

F(6)	Array of force and acceleration vectors
Y(6)	Array of position and velocity vectors

Data files: noneOutput files:

TAPE7 from subroutine SINGTJ (80 character summary)
 TAPE8 from subroutine SINGTJ (132 character line printer summary)
 TAPE16 (diagnostic output; if desired, set IERRPT to > 0)

Listing of all variables in program TJALLMAG

B	Magnitude of the magnetic field (in Gauss) [in COMMON /WRKVLU/]
BP	Value of the $B(\phi)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BR	Value of the $B(r)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BT	Value of the $B(\theta)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
CF	Character variable "F"
DELPC	Increments of rigidity spacing search in control line
DISOUT	Distance (in earth radii) for trajectory termination [in COMMON /SNGLR/]
ERAD	Average radius of the earth in kilometers [in COMMON /WRKVLU/]
ERADPL	Polar radius of the earth in km [in COMMON /GEOID/]
ERECSQ	Eccentricity of ellipsoid squared [in COMMON /GEOID/]
F(6)	Array of "F" values (velocity and acceleration in program coordinates) [in COMMON /WRKVLU/]
GCLATD	Geocentric latitude in degrees (+ = north, south = -) [in COMMON /SNGLR/]
GDAZ	Geodetic azimuth in radians (measured clockwise from north)
GDAZD	Geodetic azimuth in degrees (measured clockwise from north)
GDLATD	Geodetic latitude in degrees (+ = north, south = -) [in COMMON /SNGLR/]
GDZE	Geodetic zenith in radians (0 = vertical)
GDZED	Geodetic zenith in degrees (0 = vertical)
GLOND	Geodetic East longitude in degrees (from Greenwich meridian) [in COMMON /SNGLR/]

IDE LPC	Integer value of rigidity change increment in MV (attempt to avoid round off)
IERRPT	Integer control for printing diagnostics (normally set to 0)
INDEX	Arbitrary index number of input control line (optional)
INDO	Integer control of number of trajectories to calculate
INDXPC	Integer value of rigidity in MV increments (attempt to avoid round off)
IOSTAT	Integer system argument of status of read
IRSLT	Internal result of particle trajectory (+1 = allowed, 0 = failed, -1 = re-entrant)
ISALT	Integer value of start altitude (in km) above geoid surface
LIMIT	Limit of number of steps before trajectory is declared "Failed"
LSTEP	Number of times step size control has been reduced to overcome trajectory error
NDO	Integer control read in (number of trajectories to compute from this control line)
NTRAJC	Number of trajectories in this computer run
PC	Rigidity of particle (in units of GV)
PI	Real value of the quantity Pi (~3.1415926535) [in COMMON /TRIG/]
PIO2	Real value of Pi divided by 2.0 [in COMMON /TRIG/]
RAD	Real value of one radian (~57.29578 degrees) [in COMMON /TRIG/]
RHT	Height above geoid where a trajectory re-entered the atmosphere
RY1	Real value of the starting position of the r coordinate in r, theta, phi coordinates [in COMMON /SNGLR/]
RY2	Real value of the starting position of the theta coordinate in r, theta, phi coordinates [in COMMON /SNGLR/]
RY3	Real value of the starting position of the phi coordinate in r, theta, phi coordinates [in COMMON /SNGLR/]
SALT	Real value of the starting altitude above the surface of the geoid [in COMMON /SNGLR/]
TCD	Trigonometric cosine of the rotation angle from geodetic to geocentric
TCGDAZ	Trigonometric cosine of the geodetic azimuth (Measured clockwise from north) [in COMMON /SNGLR/]
TCGDZE	Trigonometric cosine of the geodetic zenith (measured clockwise from north)
TCY2	Trigonometric cosine of the vector theta angle in r, theta, phi coordinates [in COMMON /SNGLR/]
TCY3	Trigonometric cosine of the vector phi angle in r, theta, phi coordinates [in COMMON /SNGLR/]
TSTEP	Number steps executed in this run
TSY2	Trigonometric sine of the vector theta (theta) angle in r, theta, phi coordinates [in COMMON /SNGLR/]
TSY3	Value of the trigonometric sine of the vector phi (phi) angle in r, theta, phi coordinates [in COMMON /SNGLR/]
VEL	Particle velocity in earth radii per second [in COMMON /WRKVLU/]
Y(6)	Array of "Y" values (position and velocity in r, theta, phi coordinates) [in COMMON /WRKVLU/]
Y1GC	Starting position, r component in geocentric r, theta, phi coordinates

Y1GD	Starting position, r component in geodetic r, θ, ϕ coordinates
Y2GC	Starting position, θ component in geocentric r, θ, ϕ coordinates
Y2GD	Starting position, θ component in geodetic r, θ, ϕ coordinates
Y3GC	Starting position, ϕ component in geocentric r, θ, ϕ coordinates
Y3GD	Starting position, ϕ component in geodetic r, θ, ϕ coordinates

Subroutine GDGC (TCD, TSD)

This subroutine calculates the angle between geodetic and geocentric coordinates. The arguments TCD and TSD are the trigonometric cosine and sine of the rotation angle from a normal from the surface of the geoid (geodetic coordinates) and a radial from the center of the earth (geocentric coordinates). See Appendix B of NSSDC 72-12)

Arguments in call statement

TCD	Cosine of the rotation angle
TSD	Sine of the rotation angle

Labeled Common arguments:

Block name:	/WRKVLU/
Arguments in block	F(6), Y(6), ERAD, EOMC, VEL, BR, BT, BP
F(6)	Array of force and acceleration vectors
Y(6)	Array of position and velocity vectors
ERAD	Average radius of the earth in kilometers
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units)
VEL	Particle velocity in earth radii per second
BP	Value of the $B(\phi)$ magnetic field vector (in units of Gauss)
BR	Value of the $B(r)$ magnetic field vector (in units of Gauss)
BT	Value of the $B(\theta)$ magnetic field vector (in units of Gauss)
Block name:	/WRKTSC/
Arguments in block	TSY2, TCY2, TSY3, TCY3
TSY2	Sine of the $Y(2)$ coordinate (theta coordinate)
TCY2	Cosine of the $Y(2)$ coordinate (theta coordinate)
TSY3	Sine of the $Y(3)$ coordinate (phi coordinate)
TCY3	Cosine of the $Y(3)$ coordinate (phi coordinate)
Block name:	/TRIG/
Arguments in block	PI, RAD, PI02
PI	Value of pi
RAD	Value of degrees in a radian
PI02	Value of $\pi/2.0$
Block name:	/GEOID/
Arguments in block	ERADPL, ERECSQ
ERADPL	Polar radius of the earth in kilometers
ERECSQ	Eccentricity of ellipsoid squared
Block name:	/SNGLR/
Arguments in block	SALT, DISOUT, GCLATD, GDLATD, GLOND, GDAZD, GDZED, RY1, RY2, RY3, RHT, TSTEP
SALT	Start altitude of trajectory above surface of geoid
DISOUT	Radial distance (in earth radii) for termination of calculation
GCLATD	Geocentric latitude in degrees
GDLATD	Geodetic latitude in degrees
GLOND	East longitude in degrees

GDAZD	Geodetic azimuth in degrees
GDZED	Geodetic zenith in degrees
RY1	Original start position Y(1)
RY2	Original start position Y(2)
RY3	Original start position Y(3)
RHT	Height above geoid where trajectory re-enters the atmosphere
TSTEP	Total number of steps in this run.

Block name: /SNGLI/
 Arguments in block
 LIMIT Maximum number of steps before 'failed' trajectory
 NTRAJC Number of trajectories calculated in this run
 IERRPT Integer control for printing diagnostics (normally = 0)

Dimensioned variables: all in labeled common
 F(6) Array of force and acceleration vectors
 Y(6) Array of position and velocity vectors

Subroutines called: none

Data files: none

Output files: none

Operation:

The shape of the earth used is not a sphere, but an ellipsoid having a specified polar radius, equatorial radius, and eccentricity. When this subroutine is called, it defines the shape of an oblate earth from the polar and equatorial radius, and calculates vectors from a normal on the surface of the ellipsoid to the specified position in geodetic coordinates, at the specified latitude, and determines the vector rotation angle between geodetic coordinates and geocentric coordinates. The sine and cosine of this rotation angle are passed to the calling program. Geodetic latitude is a measure of latitude in a coordinate system normal to the surface of the earth. At a position on or above the surface of the ellipsoid, there is a slight difference between a direction normal to the surface of the ellipsoid and a direction to the geocentric. This difference is latitude dependent. (It is zero at the equator or poles and can be as large as approximately 1/2 of a degree at mid latitudes.) The vector rotation angle allows for direction specification in both geodetic (map) coordinates and geocentric coordinates. This small correction for the direction may be insignificant for some applications, but may be significant for precision calculation in a specific direction at high rigidities.

Data checking: none. The data to describe the shape of the earth are included in the subroutine.

Listing of all variables used in subroutine GDGC of program TJALLMAG

B	Magnitude of the magnetic field (in Gauss) [in COMMON /WRKVLU/]
BP	Value of the $B(\phi)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]

BR	Value of the $B(r)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BT	Value of the $B(\theta)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
DISOUT	Distance (in earth radii) for trajectory termination [in COMMON /SNGLR/]
DISTKM	Starting position geocentric radial distance from geocenter
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units) [in COMMON /WRKVLU/]
ERAD	Average radius of the earth in kilometers [in COMMON /WRKVLU/]
ERADPL	Polar radius of the earth in km [in COMMON /GEOID/]
ERECSQ	Eccentricity of ellipsoid squared [in COMMON /GEOID/]
ERPLSQ	Polar radius of earth (in km) squared
F(6)	Array of "F" values (velocity and acceleration in program coordinates) [in COMMON /WRKVLU/]
GCLATD	Geocentric latitude in degrees (+ = north, south = -) [in COMMON /SNGLR/]
GDAZD	Geodetic azimuth in degrees (measured clockwise from north)
GDCLT	Geodetic co-latitude (in radians)
GLLATD	Geodetic latitude in degrees (+ = north, south = -) [in COMMON /SNGLR/]
GDZED	Geodetic zenith in degrees (0 = vertical)
GLOND	Geodetic East longitude in degrees (from Greenwich meridian) [in COMMON /SNGLR/]
ONE	Intermediate term in computations (see NSSDC ALLMAG description)
PI	Real value of the quantity Pi (~3.1415926535) [in COMMON /TRIG/]
PIO2	Real value of Pi divided by 2.0 [in COMMON /TRIG/]
RAD	Real value of one radian (~57.29578 degrees) [in COMMON /TRIG/]
RHO	Intermediate term in computations (see NSSDC ALLMAG description)
RHT	Height above geoid where a trajectory is declared re-entrant [in COMMON /SNGLR/]
RY1	Real value of the starting position of the r coordinate in r, θ, ϕ coordinates [in COMMON /SNGLR/]
RY2	Real value of the starting position of the theta coordinate in r, θ, ϕ coordinates [in COMMON /SNGLR/]
RY3	Real value of the starting position of the phi coordinate in r, θ, ϕ coordinates [in COMMON /SNGLR/]
SALT	Real value of the starting altitude above the surface of the geoid [in COMMON /SNGLR/]
TCD	Trigonometric cosine of the rotation angle from geodetic to geocentric
TCGDCLT	Trigonometric cosine of the geocentric co-latitude
TCY2	Trigonometric cosine of the vector theta angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TCY3	Trigonometric cosine of the vector phi angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
THREE	Intermediate term in computations (see NSSDC ALLMAG description)

TSD	Trigonometric sine of the rotation angle from geodetic to geocentric
TSGDCLT	Trigonometric sine of the geocentric co-latitude
TSTEP	Number steps executed in this run
TSY2	Trigonometric sine of the vector theta (θ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TSY3	Trigonometric sine of the vector phi (ϕ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TWO	Intermediate term in computations (see NSSDC ALLMAG description)
VEL	Particle velocity in earth radii per second [in COMMON /WRKVLU/]
Y(6)	Array of "Y" values (position and velocity in r, θ, ϕ coordinates) [in COMMON /WRKVLU/]

Reference publication:

ALLMAG, GCAZLMG, LINTRA: Commuter Programs For Geomagnetic Field And Field-Line Calculations, E.G. Stassinopoulos and G.D. Mead, NSSDC 72-12, February 1972, NASA, GFSC, Greenbelt MD.

Subroutine SINGLTJ (PC, IRSLT, INDXPC, Y1GC, Y2GC, Y3GC)

This subroutine does the actual trajectory tracing. When called it initially defines control parameters and constants used in the particle tracing and initializes the Runge-Kutta variables to zero. It sets up the initial position and direction, and defines the relativistic parameters relating to the particle total energy and speed.

In this version of the subroutine, an oxygen nuclei (^{16}O) is used as the test particle. By definition a ^{16}O nuclei has a mass of 16 Atomic Mass Units (AMU) and an atomic charge of 8. The mass-energy conversion for one AMU is 0.93114 GeV. If it were desired to modify the program for some other nuclei, such as a proton that has an atomic charge of 1 and atomic mass of 1.0081415 AMU, then the rest mass energy for atomic nuclei must be adjusted.

After the initial definitions, the subroutine then chooses an initial starting step length (a relatively small value) and starts the Runge-Kutta process of tracing the particle trajectory. After each step it goes through an error checking and detection process. If the checks are satisfactory, it determines the particle location with respect to the atmosphere and the outer boundary.

If the charged particle is between the atmosphere and the outer boundary, it adjusts the size of the next step and continues the trajectory tracing until the LIMIT on the number of steps is reached.

If the charged particle is entering the atmosphere, it terminates the calculation.

If the charged particle is less than 100 km above the earth's surface, it maintains a running sum of the time at low altitudes.

If the charged particle is approaching the outer boundary, it adjusts the step size so it penetrates this boundary at small step lengths.

If the charged particle has penetrated the outer boundary at a small step, it computes the final coordinates.

When the particle has reached a solution (allowed or re-entrant) or reached the step limit, it writes out the result and returns to the calling program.

Arguments in call statement

PC, IRSLT, INDXPC, Y1GC, Y2GC, Y3GC

PC	Particle rigidity
IRSLT	Integer result of trajectory (+1 = allowed, 0 = failed, -1 = re-entrant)
INDXPC	Integer value of PC in MV units
Y1GC	Y(1) position in geocentric coordinates
Y2GC	Y(2) position in geocentric coordinates
Y3GC	Y(3) position in geocentric coordinates

Labeled Common arguments:

Block name: /WRKVLU/

Arguments in block F(6), Y(6), ERAD, EOMC, VEL, BR, BT, BP

F(6) Array of force and acceleration vectors

Y(6) Array of position and velocity vectors

ERAD Average radius of the earth in kilometers

EOMC Electronic charge divided by relativistic mass energy equivalent (mixed units)

VEL Particle velocity in earth radii per second

BP Value of the B(ϕ) magnetic field vector (in units of Gauss)

BR Value of the B(r) magnetic field vector (in units of Gauss)

BT	Value of the $B(\theta)$ magnetic field vector (in units of Gauss)
Block name:	/WRKTSC/
Arguments in block	TSY2, TCY2, TSY3, TCY3 Sine of the Y(2) coordinate (theta coordinate) Cosine of the Y(2) coordinate (theta coordinate) Sine of the Y(3) coordinate (phi coordinate) Cosine of the Y(3) coordinate (phi coordinate)
Block name:	/TRIG/
Arguments in block	PI, RAD, PI02 Value of pi Value of degrees in a radian Value of pi/2.0
Block name:	/GEOID/
Arguments in block	ERADPL, ERECSQ polar radius of the earth in kilometers Eccentricity of ellipsoid squared
Block name:	/SNGLR/
Arguments in block	SALT, DISOUT, GCLATD, GDLATD, GLOND, GDAZD, GDZED, RY1, RY2, RY3, RHT, TSTEP Start altitude of trajectory above surface of geoid Radial distance (in earth radii) for termination of calculation Geocentric latitude in degrees Geodetic latitude in degrees East longitude in degrees Geodetic azimuth in degrees Geodetic zenith in degrees Original start position Y(1), (radial component in the r, θ, ϕ coordinate system) Original start position Y(2), (theta component in the r, θ, ϕ coordinate system) Original start position Y(3), (phi component in the r, θ, ϕ coordinate system) Height above geoid where trajectory re-enters the atmosphere Total number of steps in this run.
Block name:	/SNGLI/
Arguments in block	LIMIT, NTRAJC, IERRPT Maximum number of steps before 'failed' trajectory Number of trajectories calculated in this run Integer control for printing diagnostics (normally = 0)
<u>Dimension variables:</u> (not in labeled common)	
P(6), Q(6), R(6), S(6), YB(6), FOLD(6), YOLD(6)	
P(6)	Runge-Kutta variable
Q(6)	Runge-Kutta variable
R(6)	Runge-Kutta variable
S(6)	Runge-Kutta variable
YB(6)	Runge-Kutta variable
FOLD(6)	"F" vectors of previous step
YOLD(6)	"Y" vectors of previous step

Subroutines called: FGRADAData files: noneOutput files:

TAPE7 (80 character summary)
TAPE8 (132 character line printer summary)
TAPE16 (diagnostic output; if desired set IERRPT to > 0)

Program Operation:

This program operates in the r, θ, ϕ coordinate system. The variables Y(1), Y(2), and Y(3) are the position vectors in the r, θ, ϕ coordinate system and the variables Y(4), Y(5), and Y(6) are the velocity vectors.

When this subroutine is called, it initially defines control parameters and constants used in the particle path tracing, and initializes the Runge-Kutta variables to zero. It obtains the particle's height with respect to the surface of an oblate earth. It sets up the initial position vectors, Y(1), Y(2) and Y(3), and based on the particle rigidity, sets up velocity vectors, Y(4), Y(5), and Y(6). It then defines the relativistic parameters TENG (total energy), EOMC (charge per relativistic mass/energy equivalent), and GMA (the relativistic parameter of total energy over the rest mass energy). It defines scalar quantities relating to the particle, BETA (the particle speed with respect to light), PVEL (the particle speed in earth radii per second), and HMAX (a maximum step length allowed for this particle rigidity).

Next it defines an initial starting step length (a relatively small value) and starts the Runge-Kutta process of tracing the particle trajectory. Comment cards specifically indicate the Runge-Kutta iteration process, which is the coding between FORTRAN statement numbers 130 and 170. The calls to subroutine FGRADA evaluate the $\mathbf{V} \times \mathbf{B}$ force on the particle during this step. The logic is very similar to that documented in Ralston and Wilf (1960). After each Runge-Kutta iteration step there is an extensive error checking and detection process.

The error checking process begins with a check on the particle speed (BETA), which should remain invariant throughout the trajectory. If the difference between the initial particle speed (BETA) and its current speed (RCKBETA) is greater than EDIF, then the trajectory tracing process is re-initialized (including the NSTEP variable) and the trajectory re-started at a smaller step size selection criteria. Up to five re-starts are allowed before the specific trajectory is declared impossible to calculate, evaluated as "failed", and the path length made negative in order to distinguish it from successful trajectories. In order to attempt to reach a solution the EDIF variable is widened by a factor of two after each successive trajectory failure.

After the error check, then the acceleration of the particle is compared with previous values. We have found that computational errors are most likely to occur when there are rapid changes in the acceleration. If the average change in acceleration exceeds a factor of five, or if any component of the acceleration exceeds a factor of three, then the step length for the next Runge-Kutta step is reduced.

Along the particle path the software checks the particle location with respect to the atmosphere and the outer boundary. If the charged particle is less than 100 km above the earth's surface, it maintains a running sum of the time at low altitudes. If the charged particle is entering the atmosphere, it terminates the calculation.

The next check determines if the particle has penetrated the outer termination boundary. The step length can be relatively large at extreme distances from the earth. If the outer boundary has been penetrated at a large step size, the trajectory is "backed up" and the step size reduced until it penetrates the boundary at a small step size. This results in a more precise determination of the penetration location and can significantly affect the computed asymptotic direction.

If there are no errors and the charged particle is between the atmosphere and the outer boundary, the software adjusts the size of the next step appropriate for the magnitude of the magnetic field (the step size is normally about one percent of the gyro-distance) and continues the trajectory tracing. The basic step length algorithm is:

$$H = ((2.0 * \pi * 33.333 * PC) / (B * \beta * C)) / 100.0$$

where "H" is time in seconds, "PC" is the particle rigidity in GV, "B" is the magnitude of the magnetic field in Gauss and "C" is the speed of light in km/sec.
(A handy formula to remember is the gyro-radius is 33 km per GV per Gauss)

The software initially starts at a trajectory calculation at a small step size and the step size is permitted to grow at a maximum of about 20 percent each step. If the particle trajectory starts to loop to a lower altitude, then the step size is reduced to compensate for the increasing magnitude of the magnetic field. When a loop in the trajectory develops, the acceleration forces increase and step size adjustments are made in the case of a significant increase in the acceleration forces.

During the trajectory tracing, the software notes the number of maximum and minima the trajectory experiences. This information is useful in ascertaining the complexity of the trajectory.

When the particle has reached a solution (allowed or re-entrant) or reached the step limit, the subroutine will write out the result and return to the calling program. The fate is coded in the variable IFATE: (0 = Allowed, 1 = Failed, 2 = Re-entrant, 3 = Failed, but max alt > 6.6 earth radii)

If the trajectory is allowed (penetrates the outer boundary), then the velocity vectors are transformed into asymptotic latitude and longitude. Asymptotic latitude and longitude are the geocentric coordinates the velocity vector would have at infinity. If the trajectory re-enters the atmosphere, then the position coordinates are transformed to geocentric latitude and longitude.

The 'output' of the software is in summary files, on one line for each trajectory calculation.
The output files are:

TAPE7 is in an 80-column "card image" format. This contains the initial conditions (the geodetic latitude, longitude, rigidity, zenith, and azimuth), the final results (asymptotic latitude and longitude, fate and number of steps), and a magnetic field identifier.

TAPE8 is in a line printer 132-column format. This contains more detail; the initial conditions (the geodetic latitude and the geocentric latitude, the longitude, rigidity, zenith, and azimuth), the final results (asymptotic latitude and longitude, path length,

trajectory time, time at altitudes under 100 km, number of maximum and minimum in trajectory, fate, and number of steps), and a magnetic field identifier.

TAPE16 is a diagnostic output, including a record of restarts due to BETA checks or trajectory failures.

See the examples section for samples of the output.

Possible Additions for Trajectory Plotting

If it is desired to plot a trajectory, the position variables Y(1), Y(2), and Y(3) must be stored after each Runge-Kutta step in a suitable array. The task of adding such a modification should be straightforward and is left to the individual program user.

Listing of all variables in subroutine SINGLTJ

ACCR	Magnitude of current value of the particle acceleration
ACCOLD	Magnitude of last value of the particle acceleration
AFOLD	Absolute value of FOLD
AHLT	Variable to control step size at high latitude
ANUC	Atomic number of number of nucleons in atom
ATRG1	Intermediate value to compute asymptotic latitude (avoid 0/0 problem)
ATRG2	Intermediate value to compute asymptotic latitude (avoid 0/0 problem)
AZD	Azimuth angle in degrees (measured clockwise from north)
B	Magnitude of the magnetic field (in Gauss) [in COMMON /WRKVLU/]
BETA	Particle speed as fraction of light speed ($\beta = v/c$ where c is speed of light)
BETAST	Control variable for reducing step size if error has occurred
BP	Value of the $B(\phi)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BR	Value of the $B(r)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BT	Value of the $B(\theta)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
CF	Character variable "F"
CNAME	Character variable (up to 6 characters) identifying magnetic field used
CR	Character variable "R"
DELACC	Change in particle acceleration from previous step
DISCK	Step length control variable to approach boundary at 10 % increments
DISOUT	Distance (in earth radii) for trajectory termination [in COMMON /SNGLR/]
DISTR	Radial distance (in earth radii) from current distance to termination boundary
EDIF	Variation in β allowed before error declared
EMCSQ	Mass energy equivalent for a AMU (0.931141 GeV)
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units) [in COMMON /WRKVLU/]
ERAD	Average radius of the earth in kilometers [in COMMON /WRKVLU/]
ERADPL	Polar radius of the earth in km [in COMMON /GEOID/]
ERECSQ	Eccentricity of ellipsoid squared [in COMMON /GEOID/]
F(6)	Array of "F" values (velocity and acceleration in program coordinates)

	[in COMMON /WRKVLU/]
FOLD(6)	Array of "F" values (velocity and acceleration) from previous step
FASLAT	Asymptotic latitude (in degrees)
FASLON	Asymptotic longitude (in degrees east of the Greenwich meridian)
GCLATD	Geocentric latitude in degrees (+ = north, south = -) [in COMMON /SNGLR/]
GDAZD	Geodetic azimuth in degrees (measured clockwise from north)
GDLATD	Geodetic latitude in degrees (+ = north, south = -) [in COMMON /SNGLR/]
GDZED	Geodetic zenith in degrees (0 = vertical)
GLOND	Geodetic East longitude in degrees (from Greenwich meridian)
	[in COMMON /SNGLR/]
GMA	Relativistic factor (total energy/rest energy)
GRNDKM	Altitude above surface of earth at this latitude (in km)
H	Runge-Kutta step size (in seconds)
HB	Preliminary value of step size for $\beta = 1$
HCK	Control to limit step size growth to 20%
HCNG	Change of step size from previous step
HMAX	Maximum value of step size allowed
HOLD	Value of previous Runge-Kutta step size
HSNEK	Control to approach 90% of distance to boundary
HSTART	Starting step size value (deliberately made small)
I	Index variable in do loops
IAZ	Integer value of azimuth (measured counter clockwise from north)
ICK	Index for checking acceleration growth
IERRPT	Integer control for printing diagnostics (normally set to 0)
IFATE	Integer fate of particle trajectory (0 = Allowed, 1 = Failed, 2 = Re-entrant 3 = Failed, but max alt > 6.6 earth radii)
INDXPC	Index of particle rigidity in MV
IRT	Integer control for writing results (+1 = Allowed, 0 = Failed, -1 = Re-entrant)
IRSLT	Internal result of particle trajectory (+1 = Allowed, 0 = Failed, -1 = Re-entrant)
ISALT	Integer value of start altitude (in km) above earth surface
IZE	Integer value of zenith angle (in degrees)
KBF	Number of failed attempts to trace this trajectory
LIMIT	Limit of number of steps before trajectory declared "Failed"
LSTEP	Number of times the step size control reduced to overcome trajectory error
NMAX	Number of maxima in complex trajectory path
NMIN	Number of minima in complex trajectory path
NSTEP	Number of steps in current trajectory
NSTEPT	Temporary variable that can be used to print out first 1000 steps
NTRAJC	Number of trajectories in this computer run
P(6)	Array of intermediate values used in Runge-Kutta integration
PATH	Total distance of trajectory path from start to termination
PC	Rigidity of particle (in units of GV)
PI	Real value of the quantity Pi (~3.1415926535) [in COMMON /TRIG/]
PIO2	Real value of Pi divided by 2.0 [in COMMON /TRIG/]

PSALT	Current particle distance from ground (used for re-entrant calculations)
PTCY2	Absolute value of cosine Y(2) (used in control of polar step size)
PVEL	Particle velocity (in earth radii per second)
R(6)	Array of intermediate values used in Runge-Kutta integration
R100KM	Y(1) distance of 100 km altitude at this latitude
R120KM	Y(1) distance of 120 km altitude at this latitude
RAD	Real value of one radian (~57.29578 degrees) [in COMMON /TRIG/]
RC1O6	Constant in Runge-Kutta integration (1.0/6.0)
RCKBETA	Current value of particle β after this step
RENLAT	Latitude of re-entrant particle intersection with atmosphere
RENLON	Longitude of re-entrant particle intersection with atmosphere
RFA	Ratio of acceleration magnitude between current step and last step
RFCK	Ratio of acceleration component between current step and last step
RHT	Height above geoid where a trajectory re-entered the atmosphere
RY1	Real value of the starting position of the r coordinate in r, θ, ϕ coordinates [in COMMON /SNGLR/]
RY2	Real value of the starting position of the theta coordinate in r, θ, ϕ coordinates [in COMMON /SNGLR/]
RY3	Real value of the starting position of the phi coordinate in r, θ, ϕ coordinates [in COMMON /SNGLR/]
S(6)	Array of intermediate values used in Runge-Kutta integration
SALT	Real value of the starting altitude above the surface of the geoid [in COMMON /SNGLR/]
SR2	Runge-Kutta constant (square root of 2.0)
TAU	Time (in seconds) for a trajectory transit from start to termination
TBETA	Difference between current value of β and starting value of β
TCY2	Trigonometric cosine of the vector theta (θ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TCY3	Trigonometric cosine of the vector phi (ϕ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TENG	Total energy of particle (kinetic energy plus rest mass energy)
TMS2O2	Runge-Kutta constant (2.0 - SR2/2.0)
TPS2O2	Runge-Kutta constant ((2.0 + SR2/2.0))
TSTEP	Number steps executed in this run
TSY2	Trigonometric sine of the vector theta (θ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TSY2SQ	Square of TSY2
TSY3	Value of the trigonometric sine of the vector phi (ϕ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TU100	Time (in seconds) the particle is under 100 km altitude
VEL	Particle velocity in earth radii per second [in COMMON /WRKVLU/]
Y(6)	Array of "Y" values (position and velocity in r, θ, ϕ coordinates) [in COMMON /WRKVLU/]
YB(6)	Array of intermediate values used in Runge-Kutta integration
Y10	Y(1) radial coordinate for re-entrant distance at this latitude

Y1GC	Starting position r component in geocentric coordinates
Y2GC	Starting position θ component in geocentric coordinates
Y3GC	Starting position ϕ component in geocentric coordinates
YDA5	Intermediate value for computing asymptotic latitude
YMAX	Maximum radial distance attained by trajectory
ZCHARGE	Atomic charge number
ZED	Zenith angle in degrees

Reference publication:

Ralston, A, and Wilf, S.H., Mathematical Models for Digital Computers, John Wiley and Sons, New York, 1960.

Subroutine FGRADA

This subroutine calculates the $\mathbf{V} \times \mathbf{B}$ force on the charged particle. This is a slight modification of the standard FGRAD subroutine adapted to call subroutine ALLMAG to calculate the magnetic field vectors. The version of ALLMAG included in this software is an update from the earlier version in the reference.

Arguments in call statement: none. All in labeled common

Labeled Common arguments:

Block name:	/WRKVLU/
Arguments in block	F(6), Y(6), ERAD, EOMC, VEL, BR, BT, BP
F(6)	Array of force and acceleration vectors
Y(6)	Array of position and velocity vectors
ERAD	Average radius of the earth in kilometers
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units)
VEL	Particle velocity in earth radii per second
BP	Value of the $B(\phi)$ magnetic field vector (in units of Gauss)
BR	Value of the $B(r)$ magnetic field vector (in units of Gauss)
BT	Value of the $B(\theta)$ magnetic field vector (in units of Gauss)
Block name:	/WRKTSC/
Arguments in block	TSY2, TCY2, TSY3, TCY3
TSY2	Sine of the $Y(2)$ coordinate (theta coordinate)
TCY2	Cosine of the $Y(2)$ coordinate (theta coordinate)
TSY3	Sine of the $Y(3)$ coordinate (phi coordinate)
TCY3	Cosine of the $Y(3)$ coordinate (phi coordinate)

Subroutines called:

ALLMAG (MODEL,TM,RKM,TSY2,TCY2,TSY3,TCY3,BR,BT,BP,B)	
MODEL	Integer designating the magnetic field model (14 = IGRF95)
TM	Decimal value designating the epoch of the magnetic field year (including decimal fraction of year)
RKM	Radial distance of particle from geocenter in km.
TSY2	Sine of the theta (θ) coordinate
TCY2	Cosine of the theta (θ) coordinate
TSY3	Sine of the phi (ϕ) coordinate
TCY3	Cosine of the phi (ϕ) coordinate
BR	Value of the $B(r)$ magnetic field vector (in units of Gauss)
BT	Value of the $B(\theta)$ magnetic field vector (in units of Gauss)
BP	Value of the $B(\phi)$ magnetic field vector (in units of Gauss)
B	Magnitude of the magnetic field (in Gauss)

Data files: none

Output files: none

Program Operation:

When this subroutine is called, the force vectors, ((F(1), F(2), F(3)) are defined. Then for compatibility with the standard ALLMAG call statement the epoch of the magnetic field and the magnetic field model are specified, the sine and cosine of the Y(2) and Y(3) coordinates (the θ and ϕ coordinates) are calculated, and the radial distance (in km) determined. The call to subroutine ALLMAG obtains the magnetic field vectors (BR, BT, BP) at the particle position. Then the acceleration vectors (F(4), F(5), F(6)) are calculated.

Listing of all variables in subroutine FGRADA in program TJALLMAG

B	Magnitude of the magnetic field (in Gauss) [in COMMON /WRKVLU/]
BP	Value of the $B(\phi)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BR	Value of the $B(r)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BT	Value of the $B(\theta)$ magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
EOMC	Electronic charge divided by relativistic mass energy equivalent (mixed units) [in COMMON /WRKVLU/]
ERAD	Average radius of the earth in kilometers [in COMMON /WRKVLU/]
F(6)	Array of "F" values (velocity and acceleration in program coordinates) [in COMMON /WRKVLU/]
MODEL	Integer designating the magnetic field model used in subroutine ALLMAG (needed for compatibility with standard call to subroutine ALLMAG)
RKM	Radial distance of particle from geocenter in km. (needed for compatibility with standard call to subroutine ALLMAG)
SQY6	Intermediate term $(Y6)^*Y(6)/Y(1)$
TAY2	Intermediate term TSY2/TSC2
TCY2	Trigonometric cosine of the vector theta angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TCY3	Trigonometric cosine of the vector phi angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TM	Decimal value designating the epoch of the magnetic field year (including decimal fraction of year)
TSY2	Trigonometric sine of the vector theta (θ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
TSY3	Trigonometric sine of the vector phi (ϕ) angle in r, θ, ϕ coordinates [in COMMON /SNGLR/]
VEL	Particle velocity in earth radii per second [in COMMON /WRKVLU/]
Y(6)	Array of "Y" values (position and velocity in r, θ, ϕ coordinates) [in COMMON /WRKVLU/]

Y5OY1

Intermediate term Y(5)/Y(1)

Reference publication:

ALLMAG, GCAZLMG, LINTRA: Commuter Programs For Geomagnetic Field And Field-Line Calculations, E.G. Stassinopoulos and G.D. Mead, NSSDC 72-12, February 1972, NASA, GFSC, Greenbelt MD.

Subroutine ALLMAG (MODEL,TM,RKM,ST,CT,SPH,CPH,BR,BT,BP,B)

This magnetic field subroutine is the NASA NSSDC ALLMAG software which has the option of using a number of magnetic field models. This version sets the variable MODEL to 14, which specifies the IGRF95 model of the earth's magnetic field. The variable TM specifies the epoch of the magnetic field expansion. *The FORTRAN coding in this subroutine is unchanged from that received from NASA.* The use of this NASA version of ALLMAG results in a slight deviation from the convention used in the FORTRAN software coding used elsewhere in this software assembly. In the previous subroutines, variables beginning with the character "C" are used exclusively to denote character variables. In this subroutine variables beginning with the character "C" are used, primarily to denote cosine and co-latitude, and also intermediate terms in the computation. In this subroutine the array "G"(13,13) has an equivalence set of variables and can be addressed as either real or integer.

When this subroutine is called, it first determines the model to be used and the epoch of the model to be evaluated. It then derives the magnetic field coefficients and normalizes the selected model coefficient for fast computation. Then it loads the normalized and ordered coefficients into the G arrays for magnetic field evaluation and initiates a serial computation where each term is the derivative of the previous term to evaluate the magnetic field model at the specified position.

This technique is about 30 percent slower in execution speed than the streamlined model specific code used in the TJI95 program, but is much more versatile in the ability to select different magnetic field models for any specified epoch.

Arguments in call statement:

MODEL	Integer designating the magnetic field model (14 = IGRF95)
TM	Decimal value designating the epoch of the magnetic field year (including decimal fraction of year)
RKM	Radial distance of particle from geocenter in km.
ST	Sine of the theta (θ) coordinate
CT	Cosine of the theta (θ) coordinate
SPH	Sine of the phi (ϕ) coordinate
CPH	Cosine of the phi (ϕ) coordinate
BR	Value of the $B(r)$ magnetic field vector (in units of Gauss)
BT	Value of the $B(\theta)$ magnetic field vector (in units of Gauss)
BP	Value of the $B(\phi)$ magnetic field vector (in units of Gauss)
B	Magnitude of the magnetic field (in Gauss)

Labeled Common arguments: (designed to be used with other software)

Block name:	/TRAJAC/
Arguments in block	CONSTEM, T, FILENAM
Block name:	/DIPOLE/
Arguments in block	WLONG, COLAT, EM

Dimensioned variables:

TO(14), NMX(14), ISUM(14,3), G(13,13) LSUM(14,3),

TO(14)	Array of the epoch years for the various magnetic field models
NMX(14)	The order of the field expansion for each field model
ISUM(14,3)	Check of the coefficients in data statement
G(11,11)	Normalized coefficients ordered for fast serial computation
LSUM(14,3)	Check sums for the various models

Note that the following variables are REAL*4:

GG(13,13,14), GGT(13,13,14), GGTT(13,13,14), SHMIT(13,13)

Note that the following variables are INTEGER*4:

G1(13,13), GT1(13,13), GTT1(13,13)
 G2(13,13), GT2(13,13), GTT2(13,13)
 G3(13,13), GT3(13,13), GTT3(13,13)
 G4(13,13), GT4(13,13), GTT4(13,13)
 G5(13,13), GT5(13,13), GTT5(13,13)
 G6(13,13), GT6(13,13), GTT6(13,13)
 G7(13,13), GT7(13,13), GTT7(13,13)
 G8(13,13), GT8(13,13), GTT8(13,13)
 G9(13,13), GT9(13,13), GTT9(13,13)
 G10(13,13), GT10(13,13), GTT10(13,13)
 G11(13,13), GT11(13,13), GTT11(13,13)
 G12(13,13), GT12(13,13), GTT12(13,13)
 G13(13,13), GT13(13,13), GTT13(13,13)
 G14(13,13), GT14(13,13), GTT14(13,13)
 LG(13,13,14), LGT(13,13,14), LGTT(13,13,14)

Word of caution. We have found that on some of the old SGI machines, the FORTRAN compile cannot accept the equivalence statement used in this subroutine.

Program Operation:

This subroutine is designed for computation of the earth's main magnetic field. This procedure expands the terms into FORTRAN coding (resulting in pages and pages of FORTRAN code) and then evaluates the normalized field coefficients. The result of this serial expansion is much faster than the recursion method which is much more compact in program size but requires the expansion of the Legender polynomials each time the subroutine is called.

When this subroutine is called, it first determines the model to be used and the epoch of the model to be evaluated. It then derives the magnetic field coefficients from the specified data statement and adjusts the epoch year by the time derivative in the data statements and generates normalized, ordered coefficients. (A byproduct of the magnetic field evaluation is the determination of the dipole magnitude, the position of the north magnetic dipole axis, and the offset of the dipole position from the geocenter.) Then it loads the normalized and ordered coefficients into the G arrays for magnetic field evaluation. It then determines the value of the AR variable; AR is reciprocal of the radial distance in earth radii. Each order of the field expansion requires an evaluation of AR^N where N is the order of the field expansion. (The N = 1 term, a description of a monopole magnetic field, is zero.) The first magnetic field evaluation designated by N = 2 is the dipole field component. Each subsequent order of expansion evaluates the contribution of the next order and adds this to the contribution of the previous orders. The final computation converts the magnetic field vectors to units of Gauss. (One Gauss = 10^5 Nt).

See the publication ALLMAG, GCAZLMG, LINTRA: Commuter Programs For Geomagnetic Field And Field-Line Calculations, E.G. Stassinopoulos and G.D. Mead, NSSDC 72-12, February 1972, NASA, GFSC, Greenbelt MD, for more details.

Magnetic Field Models

Model #	Identification	GSFC	99-TERM	9/65	EPOCH 1960
1	HENDRICKS & CAIN	GSFC	120-TERM	12/6	EPOCH 1960
2	CAIN ET AL.	POGO	143-TERM	10/68	EPOCH 1960
3	CAIN & LANGE	POGO	120-TERM	8/69	EPOCH 1960
4	CAIN + SWEENEY		80-TERM	10/68	EPOCH 1965
5	IGRF 1965.0		80-TERM		EPOCH 1965
6	LEATON, MALIN & EVANS, 1965		168-TERM		EPOCH 1970
7	HURWITZ (US COAST & GEODETIC)		168-TERM		EPOCH 1980
8	IGRF 1980		168-TERM		EPOCH 1975
9	IGRF 1975		80-TERM		EPOCH 1975
10	BARRACLOUGH		168-TERM		EPOCH 1975
11	AWC		168-TERM		EPOCH 1985
12	IGRF 1985		168-TERM		EPOCH 1990
13	IGRF 1990		168-TERM		EPOCH 1995
14	IGRF 1995		168-TERM		

Listing of variables in subroutine ALLMAG

AOR	Temporary value of radial distance (in earth radii) to the N th power
AR	1.0/Radial distance (in earth radii)
B	Magnitude of the magnetic field (in Gauss) [in COMMON /WRKVLU/]
BP	Value of the B(ϕ) magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BR	Value of the B(r) magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
BT	Value of the B(θ) magnetic field vector (in Gauss) [in COMMON /WRKVLU/]
C10	Intermediate value (N=10) in magnetic field expansion
C11	Intermediate value (N=11) in magnetic field expansion
C12	Intermediate value (N=12) in magnetic field expansion
C13	Intermediate value (N=13) in magnetic field expansion
C2	Intermediate value (N=2) in magnetic field expansion
C3	Intermediate value (N=3) in magnetic field expansion
C4	Intermediate value (N=4) in magnetic field expansion
C5	Intermediate value (N=5) in magnetic field expansion
C6	Intermediate value (N=6) in magnetic field expansion
C7	Intermediate value (N=7) in magnetic field expansion
C8	Intermediate value (N=8) in magnetic field expansion
C9	Intermediate value (N=9) in magnetic field expansion
COLAT	Co-latitude of the north dipole axis
CONSTEM	Magnitude of the dipole term
CP10	Trigonometric cosine function for the P10 term
CP11	Trigonometric cosine function for the P11 term
CP12	Trigonometric cosine function for the P12 term

CP13	Trigonometric cosine function for the P13 term
CP2	Trigonometric cosine function for the P2 term
CP3	Trigonometric cosine function for the P3 term
CP4	Trigonometric cosine function for the P4 term
CP5	Trigonometric cosine function for the P5 term
CP6	Trigonometric cosine function for the P6 term
CP7	Trigonometric cosine function for the P7 term
CP8	Trigonometric cosine function for the P8 term
CP9	Trigonometric cosine function for the P9 term
CP	Cosine of the phi coordinate
CT	Cosine of the theta coordinate
DP101	Derivative of the polynomial (10,1) term
DP1010	Derivative of the polynomial (10,10) term
DP102	Derivative of the polynomial (10,2) term
DP103	Derivative of the polynomial (10,3) term
DP104	Derivative of the polynomial (10,4) term
DP105	Derivative of the polynomial (10,5) term
DP106	Derivative of the polynomial (10,6) term
DP107	Derivative of the polynomial (10,7) term
DP108	Derivative of the polynomial (10,8) term
DP109	Derivative of the polynomial (10,9) term
DP111	Derivative of the polynomial (11,1) term
DP1110	Derivative of the polynomial (11,10) term
DP1111	Derivative of the polynomial (11,11) term
DP112	Derivative of the polynomial (11,2) term
DP113	Derivative of the polynomial (11,3) term
DP114	Derivative of the polynomial (11,4) term
DP115	Derivative of the polynomial (11,5) term
DP116	Derivative of the polynomial (11,6) term
DP117	Derivative of the polynomial (11,7) term
DP118	Derivative of the polynomial (11,8) term
DP119	Derivative of the polynomial (11,9) term
DP21	Derivative of the polynomial (2,1) term
DP22	Derivative of the polynomial (2,2) term
DP31	Derivative of the polynomial (3,1) term
DP32	Derivative of the polynomial (3,2) term
DP33	Derivative of the polynomial (3,3) term
DP41	Derivative of the polynomial (4,1) term
DP42	Derivative of the polynomial (4,2) term
DP43	Derivative of the polynomial (4,3) term
DP44	Derivative of the polynomial (4,4) term
DP51	Derivative of the polynomial (5,1) term
DP52	Derivative of the polynomial (5,2) term
DP53	Derivative of the polynomial (5,3) term
DP54	Derivative of the polynomial (5,4) term
DP55	Derivative of the polynomial (5,5) term
DP61	Derivative of the polynomial (6,1) term
DP62	Derivative of the polynomial (6,2) term
DP63	Derivative of the polynomial (6,3) term

DP64	Derivative of the polynomial (6,4) term
DP65	Derivative of the polynomial (6,5) term
DP66	Derivative of the polynomial (6,6) term
DP71	Derivative of the polynomial (7,1) term
DP72	Derivative of the polynomial (7,2) term
DP73	Derivative of the polynomial (7,3) term
DP74	Derivative of the polynomial (7,4) term
DP75	Derivative of the polynomial (7,5) term
DP76	Derivative of the polynomial (7,6) term
DP77	Derivative of the polynomial (7,7) term
DP81	Derivative of the polynomial (8,1) term
DP82	Derivative of the polynomial (8,2) term
DP83	Derivative of the polynomial (8,3) term
DP84	Derivative of the polynomial (8,4) term
DP85	Derivative of the polynomial (8,5) term
DP86	Derivative of the polynomial (8,6) term
DP87	Derivative of the polynomial (8,7) term
DP88	Derivative of the polynomial (8,8) term
DP91	Derivative of the polynomial (9,1) term
DP92	Derivative of the polynomial (9,2) term
DP93	Derivative of the polynomial (9,3) term
DP94	Derivative of the polynomial (9,4) term
DP95	Derivative of the polynomial (9,5) term
DP96	Derivative of the polynomial (9,6) term
DP97	Derivative of the polynomial (9,7) term
DP98	Derivative of the polynomial (9,8) term
DP99	Derivative of the polynomial (9,9) term
FILENAM	Externally provided identification
G(13,13)	Array of normalized magnetic field coefficients ordered for fast computation
I	Index in do loops
JJ	Integer control parameter for loading coefficients
K	Index in do loops
L	Intermediate index for data checking
M	Intermediate index for data checking
MODOLD	Integer control variable for model changing
N	Index in specifying degree and order of magnetic field expansion
P101	Polynomial (10,1) term
P1010	Polynomial (10,10) term
P102	Polynomial (10,2) term
P103	Polynomial (10,3) term
P104	Polynomial (10,4) term
P105	Polynomial (10,5) term

P106	Polynomial (10,6) term
P107	Polynomial (10,7) term
P108	Polynomial (10,8) term
P109	Polynomial (10,9) term
P111	Polynomial (11,1) term
P110	Polynomial (11,10) term
P1111	Polynomial (11,11) term
P112	Polynomial (11,2) term
P113	Polynomial (11,3) term
P114	Polynomial (11,4) term
P115	Polynomial (11,5) term
P116	Polynomial (11,6) term
P117	Polynomial (11,7) term
P118	Polynomial (11,8) term
P119	Polynomial (11,9) term
P21	Polynomial (2,1) term
P22	Polynomial (2,2) term
P31	Polynomial (3,1) term
P32	Polynomial (3,2) term
P33	Polynomial (3,3) term
P41	Polynomial (4,1) term
P42	Polynomial (4,2) term
P43	Polynomial (4,3) term
P44	Polynomial (4,4) term
P51	Polynomial (5,1) term
P52	Polynomial (5,2) term
P53	Polynomial (5,3) term
P54	Polynomial (5,4) term
P55	Polynomial (5,5) term
P61	Polynomial (6,1) term
P62	Polynomial (6,2) term
P63	Polynomial (6,3) term
P64	Polynomial (6,4) term
P65	Polynomial (6,5) term
P66	Polynomial (6,6) term
P71	Polynomial (7,1) term
P72	Polynomial (7,2) term
P73	Polynomial (7,3) term
P74	Polynomial (7,4) term
P75	Polynomial (7,5) term
P76	Polynomial (7,6) term
P77	Polynomial (7,7) term
P81	Polynomial (8,1) term
P82	Polynomial (8,2) term
P83	Polynomial (8,3) term
P84	Polynomial (8,4) term
P85	Polynomial (8,5) term
P86	Polynomial (8,6) term
P87	Polynomial (8,7) term
P88	Polynomial (8,8) term
P91	Polynomial (9,1) term

P92	Polynomial (9,2) term
P93	Polynomial (9,3) term
P94	Polynomial (9,4) term
P95	Polynomial (9,5) term
P96	Polynomial (9,6) term
P97	Polynomial (9,7) term
P98	Polynomial (9,8) term
P99	Polynomial (9,9) term
SHMIT	Array of coefficients from model data statements
SP10	Trigonometric sine function for the P10 term
SP11	Trigonometric sine function for the P11 term
SP2	Trigonometric sine function for the P2 term
SP3	Trigonometric sine function for the P3 term
SP4	Trigonometric sine function for the P4 term
SP5	Trigonometric sine function for the P5 term
SP6	Trigonometric sine function for the P6 term
SP7	Trigonometric sine function for the P7 term
SP8	Trigonometric sine function for the P8 term
SP9	Trigonometric sine function for the P9 term
SPH	Sine of the phi coordinate
ST	Sine of the theta coordinate
T	Difference in decimal years between specified epoch and model reference year
TO	Reference year for the specified magnetic field model
TM	Time (decimal year) for the selected magnetic field model to be evaluated.
TMOLD	Year of the previous magnetic field model used
WLONG	West longitude of dipole axis

Reference publications:

ALLMAG, GCAZLMG, LINTRA: Commuter Programs For Geomagnetic Field And Field-Line Calculations, E.G. Stassinopulous and G.D. Mead, NSSDC 72-12, February 1972, NASA, GFSC, Greenbelt MD.

Computation of the Main Geomagnetic Field for Spherical Harmonic Expansion, Data Users Note NSSDC 68-11, February 1968, NSDC, NASA, GFSC, Greenbelt MD.

Example of Program TJI95T Execution.

This example contains the following:

A TAPE1 input data file.

A RUN.DOC which illustrates the on-screen displays as the program executes.

A TAPE7 output file (sorted to remove the duplicate trajectories).

A TAPE8 output file (exactly as generated during the computer run).

This example is an illustration of an attempt to determine the vertical cutoff rigidity of a balloon launching location in the US. Specifically Palestine Texas coordinates 31.78 degrees North latitude and 264.37 East Longitude. (Remember that all longitudes are measured east from the Greenwich meridian.)

The TAPE1 data input file contains three data lines. The first data line initially specifies a search at 1 GV intervals beginning at 20 GV. The second data line specifies a search at 0.1 GV intervals beginning at 8 GV for n trajectories. The third data line specifies a search at 0.01 GV intervals beginning at 5.5 GV for n trajectories. The result of this execution generates a TAPE7 data file to be evaluated. Note that there are initially simple trajectories that complete in 65 steps. As the rigidity decreases, the number of steps increases and the first re-entrant trajectory occurs at a rigidity of 4.61 GV. Between 4.62 GV and 3.75 GV there is the complex structure of the cosmic ray penumbra with its allowed and forbidden structure. At rigidities of 3.74 GV and below, all charged particles fail to reach interplanetary space.

The summary of this run is $R_U = 4.62$ GV, $R_L = 3.75$ GV, the effective cutoff is $R_C = 4.26$ GV and the penumbra was computed to be 0.87 GV wide in the vertical direction.

This is the TAPE1 data input file.

Note the identification comments after the negative rigidity termination value.

This is the "on-line" output as the program executes

TAPE 1 31.78 264.37 20.00 0.00 0.00 1.00 16 0 1
 TAPE 1 31.78 264.37 8.00 0.00 0.00 0.10 50 0 2
 TAPE 1 31.78 264.37 5.50 0.00 0.00 0.01 250 0 3
 END OF DATA INPUT (NEGATIVE VALUE READ IN)

TOTAL NUMBER OF STEPS 311048.

TOTAL NUMBER OF TRAJECTORIES 316

End program TJI95T

This is the TAPE7 output file.

<i>Lat</i>	<i>Long</i>	<i>Rig</i>	<i>Zen</i>	<i>Az</i>	<i>Alt</i>	<i>ALat</i>	<i>ALon</i>	<i>Nstep</i>	<i>Fate</i>	<i>ID</i>
31.78	264.37	20.000	0.0	0.0	20	-5.24	313.32	65	0	I95
31.78	264.37	19.000	0.0	0.0	20	-7.71	314.82	63	0	I95
31.78	264.37	18.000	0.0	0.0	20	-10.38	316.46	65	0	I95
31.78	264.37	17.000	0.0	0.0	20	-13.23	318.28	68	0	I95
31.78	264.37	16.000	0.0	0.0	20	-16.25	320.34	71	0	I95
31.78	264.37	15.000	0.0	0.0	20	-19.38	322.70	74	0	I95
31.78	264.37	14.000	0.0	0.0	20	-22.52	325.47	77	0	I95
31.78	264.37	13.000	0.0	0.0	20	-25.48	328.74	81	0	I95
31.78	264.37	12.000	0.0	0.0	20	-27.97	332.62	86	0	I95
31.78	264.37	11.000	0.0	0.0	20	-29.57	337.09	91	0	I95
31.78	264.37	10.000	0.0	0.0	20	-29.79	342.01	98	0	I95
31.78	264.37	9.000	0.0	0.0	20	-28.23	347.18	106	0	I95
31.78	264.37	8.000	0.0	0.0	20	-24.80	353.07	115	0	I95
31.78	264.37	7.900	0.0	0.0	20	-24.36	353.77	116	0	I95
31.78	264.37	7.800	0.0	0.0	20	-23.91	354.50	118	0	I95
31.78	264.37	7.700	0.0	0.0	20	-23.43	355.27	118	0	I95
31.78	264.37	7.600	0.0	0.0	20	-22.94	356.09	119	0	I95
31.78	264.37	7.500	0.0	0.0	20	-22.43	356.95	123	0	I95
31.78	264.37	7.400	0.0	0.0	20	-21.90	357.88	124	0	I95
31.78	264.37	7.300	0.0	0.0	20	-21.36	358.87	126	0	I95
31.78	264.37	7.200	0.0	0.0	20	-20.79	359.93	125	0	I95
31.78	264.37	7.100	0.0	0.0	20	-20.19	1.07	125	0	I95
31.78	264.37	7.000	0.0	0.0	20	-19.57	2.30	127	0	I95
31.78	264.37	6.900	0.0	0.0	20	-18.91	3.63	129	0	I95
31.78	264.37	6.800	0.0	0.0	20	-18.22	5.07	129	0	I95
31.78	264.37	6.700	0.0	0.0	20	-17.48	6.63	132	0	I95
31.78	264.37	6.600	0.0	0.0	20	-16.69	8.33	132	0	I95
31.78	264.37	6.500	0.0	0.0	20	-15.83	10.18	134	0	I95
31.78	264.37	6.400	0.0	0.0	20	-14.89	12.19	136	0	I95
31.78	264.37	6.300	0.0	0.0	20	-13.85	14.38	138	0	I95
31.78	264.37	6.200	0.0	0.0	20	-12.68	16.76	141	0	I95
31.78	264.37	6.100	0.0	0.0	20	-11.35	19.37	143	0	I95
31.78	264.37	6.000	0.0	0.0	20	-9.82	22.22	143	0	I95
31.78	264.37	5.900	0.0	0.0	20	-8.05	25.33	146	0	I95
31.78	264.37	5.800	0.0	0.0	20	-5.96	28.75	148	0	I95
31.78	264.37	5.700	0.0	0.0	20	-3.50	32.51	151	0	I95
31.78	264.37	5.600	0.0	0.0	20	-0.56	36.69	155	0	I95
31.78	264.37	5.500	0.0	0.0	20	2.95	41.40	156	0	I95
31.78	264.37	5.490	0.0	0.0	20	3.33	41.91	156	0	I95
31.78	264.37	5.480	0.0	0.0	20	3.73	42.42	158	0	I95
31.78	264.37	5.470	0.0	0.0	20	4.13	42.94	158	0	I95
31.78	264.37	5.460	0.0	0.0	20	4.54	43.47	158	0	I95
31.78	264.37	5.450	0.0	0.0	20	4.95	44.01	158	0	I95
31.78	264.37	5.440	0.0	0.0	20	5.37	44.56	159	0	I95
31.78	264.37	5.430	0.0	0.0	20	5.80	45.11	159	0	I95
31.78	264.37	5.420	0.0	0.0	20	6.24	45.68	158	0	I95
31.78	264.37	5.410	0.0	0.0	20	6.68	46.25	160	0	I95
31.78	264.37	5.400	0.0	0.0	20	7.14	46.84	160	0	I95
31.78	264.37	5.390	0.0	0.0	20	7.60	47.43	161	0	I95
31.78	264.37	5.380	0.0	0.0	20	8.07	48.04	161	0	I95
31.78	264.37	5.370	0.0	0.0	20	8.54	48.65	162	0	I95
31.78	264.37	5.360	0.0	0.0	20	9.02	49.28	162	0	I95
31.78	264.37	5.350	0.0	0.0	20	9.52	49.92	162	0	I95
31.78	264.37	5.340	0.0	0.0	20	10.02	50.58	162	0	I95
31.78	264.37	5.330	0.0	0.0	20	10.52	51.25	163	0	I95
31.78	264.37	5.320	0.0	0.0	20	11.04	51.93	163	0	I95
31.78	264.37	5.310	0.0	0.0	20	11.56	52.63	165	0	I95
31.78	264.37	5.300	0.0	0.0	20	12.09	53.35	165	0	I95
31.78	264.37	5.290	0.0	0.0	20	12.63	54.08	165	0	I95
31.78	264.37	5.280	0.0	0.0	20	13.18	54.83	165	0	I95
31.78	264.37	5.270	0.0	0.0	20	13.73	55.60	166	0	I95
31.78	264.37	5.260	0.0	0.0	20	14.29	56.39	166	0	I95
31.78	264.37	5.250	0.0	0.0	20	14.86	57.21	168	0	I95

<i>Lat</i>	<i>Long</i>	<i>Rig</i>	<i>Zen</i>	<i>Az</i>	<i>Alt</i>	<i>ALat</i>	<i>ALon</i>	<i>Nstep</i>	<i>Fate</i>	<i>ID</i>
31.78	264.37	5.240	0.0	0.0	20	15.44	58.04	169	0	I95
31.78	264.37	5.230	0.0	0.0	20	16.02	58.90	169	0	I95
31.78	264.37	5.220	0.0	0.0	20	16.60	59.79	170	0	I95
31.78	264.37	5.210	0.0	0.0	20	17.20	60.70	169	0	I95
31.78	264.37	5.200	0.0	0.0	20	17.79	61.65	170	0	I95
31.78	264.37	5.190	0.0	0.0	20	18.39	62.62	170	0	I95
31.78	264.37	5.180	0.0	0.0	20	19.00	63.63	171	0	I95
31.78	264.37	5.170	0.0	0.0	20	19.61	64.67	173	0	I95
31.78	264.37	5.160	0.0	0.0	20	20.21	65.75	174	0	I95
31.78	264.37	5.150	0.0	0.0	20	20.82	66.88	173	0	I95
31.78	264.37	5.140	0.0	0.0	20	21.43	68.04	173	0	I95
31.78	264.37	5.130	0.0	0.0	20	22.03	69.25	174	0	I95
31.78	264.37	5.120	0.0	0.0	20	22.63	70.52	176	0	I95
31.78	264.37	5.110	0.0	0.0	20	23.22	71.83	177	0	I95
31.78	264.37	5.100	0.0	0.0	20	23.80	73.20	179	0	I95
31.78	264.37	5.090	0.0	0.0	20	24.37	74.63	179	0	I95
31.78	264.37	5.080	0.0	0.0	20	24.92	76.13	178	0	I95
31.78	264.37	5.070	0.0	0.0	20	25.45	77.70	178	0	I95
31.78	264.37	5.060	0.0	0.0	20	25.95	79.34	181	0	I95
31.78	264.37	5.050	0.0	0.0	20	26.43	81.06	180	0	I95
31.78	264.37	5.040	0.0	0.0	20	26.87	82.86	185	0	I95
31.78	264.37	5.030	0.0	0.0	20	27.27	84.75	184	0	I95
31.78	264.37	5.020	0.0	0.0	20	27.61	86.74	184	0	I95
31.78	264.37	5.010	0.0	0.0	20	27.90	88.82	183	0	I95
31.78	264.37	5.000	0.0	0.0	20	28.12	91.01	185	0	I95
31.78	264.37	4.990	0.0	0.0	20	28.26	93.31	187	0	I95
31.78	264.37	4.980	0.0	0.0	20	28.30	95.72	190	0	I95
31.78	264.37	4.970	0.0	0.0	20	28.23	98.25	191	0	I95
31.78	264.37	4.960	0.0	0.0	20	28.04	100.90	191	0	I95
31.78	264.37	4.950	0.0	0.0	20	27.71	103.67	192	0	I95
31.78	264.37	4.940	0.0	0.0	20	27.20	106.57	193	0	I95
31.78	264.37	4.930	0.0	0.0	20	26.51	109.60	193	0	I95
31.78	264.37	4.920	0.0	0.0	20	25.59	112.75	194	0	I95
31.78	264.37	4.910	0.0	0.0	20	24.42	116.04	196	0	I95
31.78	264.37	4.900	0.0	0.0	20	22.96	119.46	197	0	I95
31.78	264.37	4.890	0.0	0.0	20	21.17	123.03	200	0	I95
31.78	264.37	4.880	0.0	0.0	20	19.01	126.75	199	0	I95
31.78	264.37	4.870	0.0	0.0	20	16.41	130.66	199	0	I95
31.78	264.37	4.860	0.0	0.0	20	13.32	134.81	200	0	I95
31.78	264.37	4.850	0.0	0.0	20	9.66	139.29	205	0	I95
31.78	264.37	4.840	0.0	0.0	20	5.36	144.24	206	0	I95
31.78	264.37	4.830	0.0	0.0	20	0.36	149.94	207	0	I95
31.78	264.37	4.820	0.0	0.0	20	-5.35	156.93	210	0	I95
31.78	264.37	4.810	0.0	0.0	20	-11.52	166.27	215	0	I95
31.78	264.37	4.800	0.0	0.0	20	-17.06	180.21	220	0	I95
31.78	264.37	4.790	0.0	0.0	20	-17.37	203.45	226	0	I95
31.78	264.37	4.780	0.0	0.0	20	4.88	250.24	248	0	I95
31.78	264.37	4.770	0.0	0.0	20	-13.70	250.39	335	0	I95
31.78	264.37	4.760	0.0	0.0	20	-8.61	263.29	564	0	I95
31.78	264.37	4.750	0.0	0.0	20	-26.56	269.82	461	0	I95
31.78	264.37	4.740	0.0	0.0	20	12.03	481.81	960	0	I95
31.78	264.37	4.730	0.0	0.0	20	12.32	216.93	318	0	I95
31.78	264.37	4.720	0.0	0.0	20	7.00	154.00	308	0	I95
31.78	264.37	4.710	0.0	0.0	20	-5.88	162.23	312	0	I95
31.78	264.37	4.700	0.0	0.0	20	1.62	276.29	367	0	I95
31.78	264.37	4.690	0.0	0.0	20	18.22	418.43	801	0	I95
31.78	264.37	4.680	0.0	0.0	20	7.75	159.08	407	0	I95
31.78	264.37	4.670	0.0	0.0	20	6.33	196.92	439	0	I95
31.78	264.37	4.660	0.0	0.0	20	12.68	223.97	494	0	I95
31.78	264.37	4.650	0.0	0.0	20	0.48	172.80	526	0	I95
31.78	264.37	4.640	0.0	0.0	20	19.34	199.59	582	0	I95
31.78	264.37	4.630	0.0	0.0	20	9.04	185.56	634	0	I95
31.78	264.37	4.620	0.0	0.0	20	19.50	194.45	692	0	I95
31.78	264.37	4.610	0.0	0.0	20	R	R	451	1	I95
31.78	264.37	4.600	0.0	0.0	20	R	R	470	1	I95

<i>Lat</i>	<i>Long</i>	<i>Rig</i>	<i>Zen</i>	<i>Az</i>	<i>Alt</i>	<i>ALat</i>	<i>ALon</i>	<i>Nstep</i>	<i>Fate</i>	<i>ID</i>
31.78	264.37	4.590	0.0	0.0	20	R	223.09	755	0	I95
31.78	264.37	4.580	0.0	0.0	20	15.48	189.53	657	0	I95
31.78	264.37	4.570	0.0	0.0	20	14.40	160.78	621	0	I95
31.78	264.37	4.560	0.0	0.0	20	-3.44	375.39	1127	0	I95
31.78	264.37	4.550	0.0	0.0	20	4.71	155.49	550	0	I95
31.78	264.37	4.540	0.0	0.0	20	18.17	181.50	531	0	I95
31.78	264.37	4.530	0.0	0.0	20	8.38	587.32	1590	0	I95
31.78	264.37	4.520	0.0	0.0	20	12.49	226.39	494	0	I95
31.78	264.37	4.510	0.0	0.0	20	4.61	146.56	458	0	I95
31.78	264.37	4.500	0.0	0.0	20	15.27	156.54	444	0	I95
31.78	264.37	4.490	0.0	0.0	20	4.04	359.25	601	0	I95
31.78	264.37	4.480	0.0	0.0	20	R	R	922	1	I95
31.78	264.37	4.470	0.0	0.0	20	-4.98	195.20	503	0	I95
31.78	264.37	4.460	0.0	0.0	20	-8.20	391.59	834	0	I95
31.78	264.37	4.450	0.0	0.0	20	-17.87	271.53	416	0	I95
31.78	264.37	4.440	0.0	0.0	20	8.38	166.72	377	0	I95
31.78	264.37	4.430	0.0	0.0	20	4.88	145.24	367	0	I95
31.78	264.37	4.420	0.0	0.0	20	7.96	140.45	363	0	I95
31.78	264.37	4.410	0.0	0.0	20	12.08	149.30	364	0	I95
31.78	264.37	4.400	0.0	0.0	20	8.32	179.00	370	0	I95
31.78	264.37	4.390	0.0	0.0	20	14.63	399.09	833	0	I95
31.78	264.37	4.380	0.0	0.0	20	2.01	653.07	1474	0	I95
31.78	264.37	4.370	0.0	0.0	20	R	R	1098	1	I95
31.78	264.37	4.360	0.0	0.0	20	3.30	218.79	527	0	I95
31.78	264.37	4.350	0.0	0.0	20	-19.57	329.38	670	0	I95
31.78	264.37	4.340	0.0	0.0	20	-6.96	213.68	456	0	I95
31.78	264.37	4.330	0.0	0.0	20	R	R	3719	1	I95
31.78	264.37	4.320	0.0	0.0	20	R	R	705	1	I95
31.78	264.37	4.310	0.0	0.0	20	R	R	1180	1	I95
31.78	264.37	4.300	0.0	0.0	20	-8.79	317.03	816	0	I95
31.78	264.37	4.290	0.0	0.0	20	R	R	417	1	I95
31.78	264.37	4.280	0.0	0.0	20	R	R	410	1	I95
31.78	264.37	4.270	0.0	0.0	20	R	R	394	1	I95
31.78	264.37	4.260	0.0	0.0	20	R	R	393	1	I95
31.78	264.37	4.250	0.0	0.0	20	R	R	396	1	I95
31.78	264.37	4.240	0.0	0.0	20	R	R	410	1	I95
31.78	264.37	4.230	0.0	0.0	20	R	R	432	1	I95
31.78	264.37	4.220	0.0	0.0	20	R	R	518	1	I95
31.78	264.37	4.210	0.0	0.0	20	R	R	767	0	I95
31.78	264.37	4.200	0.0	0.0	20	-9.19	317.95	758	1	I95
31.78	264.37	4.190	0.0	0.0	20	R	R	960	1	I95
31.78	264.37	4.180	0.0	0.0	20	R	R	672	0	I95
31.78	264.37	4.170	0.0	0.0	20	11.88	442.61	1227	0	I95
31.78	264.37	4.160	0.0	0.0	20	-5.98	534.71	1643	0	I95
31.78	264.37	4.150	0.0	0.0	20	22.85	791.83	1277	0	I95
31.78	264.37	4.140	0.0	0.0	20	-4.83	653.54	3366	1	I95
31.78	264.37	4.130	0.0	0.0	20	R	R	2574	1	I95
31.78	264.37	4.120	0.0	0.0	20	R	R	2497	1	I95
31.78	264.37	4.110	0.0	0.0	20	R	R	739	1	I95
31.78	264.37	4.100	0.0	0.0	20	R	R	2871	0	I95
31.78	264.37	4.090	0.0	0.0	20	-5.46	912.74	717	1	I95
31.78	264.37	4.080	0.0	0.0	20	R	R	708	1	I95
31.78	264.37	4.070	0.0	0.0	20	R	R	1221	0	I95
31.78	264.37	4.060	0.0	0.0	20	3.50	515.42	1272	1	I95
31.78	264.37	4.050	0.0	0.0	20	R	R	1859	1	I95
31.78	264.37	4.040	0.0	0.0	20	R	R	2069	1	I95
31.78	264.37	4.030	0.0	0.0	20	R	R	1388	1	I95
31.78	264.37	4.020	0.0	0.0	20	R	R	682	1	I95
31.78	264.37	4.010	0.0	0.0	20	R	R	725	1	I95
31.78	264.37	4.000	0.0	0.0	20	R	R	1426	1	I95
31.78	264.37	3.990	0.0	0.0	20	R	R	1400	1	I95
31.78	264.37	3.980	0.0	0.0	20	R	R	29.93	0	I95
31.78	264.37	3.970	0.0	0.0	20	13.24	1170.28	3323	0	I95
31.78	264.37	3.960	0.0	0.0	20	3.76	832.00	3286	0	I95
31.78	264.37	3.950	0.0	0.0	20	R	R	1887	0	I95
31.78	264.37	3.940	0.0	0.0	20	R	R	1350	1	I95
31.78	264.37	3.930	0.0	0.0	20	R	R	669	1	I95
31.78	264.37	3.920	0.0	0.0	20	R	R	581	1	I95
31.78	264.37	3.910	0.0	0.0	20	R	R	3662	1	I95

31.78	264.37	3.900	0.0	0.0	20	R	R	1151	1	I95
31.78	264.37	3.890	0.0	0.0	20	R	R	2101	1	I95
31.78	264.37	3.880	0.0	0.0	20	R	R	1214	1	I95
31.78	264.37	3.870	0.0	0.0	20	R	R	832	1	I95
31.78	264.37	3.860	0.0	0.0	20	R	R	4138	1	I95
31.78	264.37	3.850	0.0	0.0	20	R	R	808	1	I95
31.78	264.37	3.840	0.0	0.0	20	R	R	1454	1	I95
31.78	264.37	3.830	0.0	0.0	20	R	R	681	1	I95
31.78	264.37	3.820	0.0	0.0	20	R	R	598	1	I95
31.78	264.37	3.810	0.0	0.0	20	R	R	1344	1	I95
31.78	264.37	3.800	0.0	0.0	20	R	R	1605	1	I95
31.78	264.37	3.790	0.0	0.0	20	R	R	1737	1	I95
31.78	264.37	3.780	0.0	0.0	20	R	R	3492	1	I95
31.78	264.37	3.770	0.0	0.0	20	R	R	1055	1	I95
31.78	264.37	3.760	0.0	0.0	20	R	R	2490	1	I95
31.78	264.37	3.750	0.0	0.0	20	-6.45	635.61	1843	0	I95
31.78	264.37	3.740	0.0	0.0	20	R	R	935	1	I95
31.78	264.37	3.730	0.0	0.0	20	R	R	1911	1	I95
31.78	264.37	3.720	0.0	0.0	20	R	R	1446	1	I95
31.78	264.37	3.710	0.0	0.0	20	R	R	2609	1	I95
31.78	264.37	3.700	0.0	0.0	20	R	R	2158	1	I95
31.78	264.37	3.690	0.0	0.0	20	R	R	3278	1	I95
31.78	264.37	3.680	0.0	0.0	20	R	R	3106	1	I95
31.78	264.37	3.670	0.0	0.0	20	R	R	651	1	I95
31.78	264.37	3.660	0.0	0.0	20	R	R	568	1	I95
31.78	264.37	3.650	0.0	0.0	20	R	R	547	1	I95
31.78	264.37	3.640	0.0	0.0	20	R	R	526	1	I95
31.78	264.37	3.630	0.0	0.0	20	R	R	506	1	I95
31.78	264.37	3.620	0.0	0.0	20	R	R	498	1	I95
31.78	264.37	3.610	0.0	0.0	20	R	R	478	1	I95
31.78	264.37	3.600	0.0	0.0	20	R	R	480	1	I95
31.78	264.37	3.590	0.0	0.0	20	R	R	477	1	I95
31.78	264.37	3.580	0.0	0.0	20	R	R	482	1	I95
31.78	264.37	3.570	0.0	0.0	20	R	R	484	1	I95
31.78	264.37	3.560	0.0	0.0	20	R	R	500	1	I95
31.78	264.37	3.550	0.0	0.0	20	R	R	532	1	I95
31.78	264.37	3.540	0.0	0.0	20	R	R	542	1	I95
31.78	264.37	3.530	0.0	0.0	20	R	R	591	1	I95
31.78	264.37	3.520	0.0	0.0	20	R	R	618	1	I95
31.78	264.37	3.510	0.0	0.0	20	R	R	13124	1	I95
31.78	264.37	3.500	0.0	0.0	20	R	R	1899	1	I95
31.78	264.37	3.490	0.0	0.0	20	R	R	1472	1	I95
31.78	264.37	3.480	0.0	0.0	20	R	R	1650	1	I95
31.78	264.37	3.470	0.0	0.0	20	R	R	2147	1	I95
31.78	264.37	3.460	0.0	0.0	20	R	R	738	1	I95
31.78	264.37	3.450	0.0	0.0	20	R	R	2220	1	I95
31.78	264.37	3.440	0.0	0.0	20	R	R	1831	1	I95
31.78	264.37	3.430	0.0	0.0	20	R	R	1000	1	I95
31.78	264.37	3.420	0.0	0.0	20	R	R	3053	1	I95
31.78	264.37	3.410	0.0	0.0	20	R	R	614	1	I95
31.78	264.37	3.400	0.0	0.0	20	R	R	3796	1	I95
31.78	264.37	3.390	0.0	0.0	20	R	R	1897	1	I95
31.78	264.37	3.380	0.0	0.0	20	R	R	4799	1	I95
31.78	264.37	3.370	0.0	0.0	20	R	R	2721	1	I95
31.78	264.37	3.360	0.0	0.0	20	R	R	2217	1	I95
31.78	264.37	3.350	0.0	0.0	20	R	R	840	1	I95
31.78	264.37	3.340	0.0	0.0	20	R	R	2002	1	I95
31.78	264.37	3.330	0.0	0.0	20	R	R	846	1	I95
31.78	264.37	3.320	0.0	0.0	20	R	R	2328	1	I95
31.78	264.37	3.310	0.0	0.0	20	R	R	1881	1	I95
31.78	264.37	3.300	0.0	0.0	20	R	R	5059	1	I95
31.78	264.37	3.290	0.0	0.0	20	R	R	2375	1	I95
31.78	264.37	3.280	0.0	0.0	20	R	R	1039	1	I95
31.78	264.37	3.270	0.0	0.0	20	R	R	2507	1	I95
31.78	264.37	3.260	0.0	0.0	20	R	R	1101	1	I95
31.78	264.37	3.250	0.0	0.0	20	R	R	2663	1	I95

<i>Lat</i>	<i>Long</i>	<i>Rig</i>	<i>Zen</i>	<i>Az</i>	<i>Alt</i>	<i>ALat</i>	<i>ALon</i>	<i>Nstep</i>	<i>Fate</i>	<i>ID</i>
31.78	264.37	3.240	0.0	0.0	20	R	R	1368	1	I95
31.78	264.37	3.230	0.0	0.0	20	R	R	1561	1	I95
31.78	264.37	3.220	0.0	0.0	20	R	R	679	1	I95
31.78	264.37	3.210	0.0	0.0	20	R	R	602	1	I95
31.78	264.37	3.200	0.0	0.0	20	R	R	603	1	I95
31.78	264.37	3.190	0.0	0.0	20	R	R	687	1	I95
31.78	264.37	3.180	0.0	0.0	20	R	R	1569	1	I95
31.78	264.37	3.170	0.0	0.0	20	R	R	917	1	I95
31.78	264.37	3.160	0.0	0.0	20	R	R	825	1	I95
31.78	264.37	3.150	0.0	0.0	20	R	R	2100	1	I95
31.78	264.37	3.140	0.0	0.0	20	R	R	785	1	I95
31.78	264.37	3.130	0.0	0.0	20	R	R	4517	1	I95
31.78	264.37	3.120	0.0	0.0	20	R	R	2232	1	I95
31.78	264.37	3.110	0.0	0.0	20	R	R	3151	1	I95
31.78	264.37	3.100	0.0	0.0	20	R	R	3059	1	I95
31.78	264.37	3.090	0.0	0.0	20	R	R	2487	1	I95
31.78	264.37	3.080	0.0	0.0	20	R	R	2140	1	I95
31.78	264.37	3.070	0.0	0.0	20	R	R	736	1	I95
31.78	264.37	3.060	0.0	0.0	20	R	R	705	1	I95
31.78	264.37	3.050	0.0	0.0	20	R	R	848	1	I95
31.78	264.37	3.040	0.0	0.0	20	R	R	1452	1	I95
31.78	264.37	3.030	0.0	0.0	20	R	R	2468	1	I95
31.78	264.37	3.020	0.0	0.0	20	R	R	2662	1	I95
31.78	264.37	3.010	0.0	0.0	20	R	R	5218	1	I95

This is the TAPE8 output file.

1 RUN START DATE 2000/12/23@22: 0:37

	<i>GD Lat</i>	<i>GC Lat</i>	<i>E Long</i>	<i>Ze</i>	<i>Az</i>	<i>Rig</i>	<i>Alon</i>	<i>Alat</i>	<i>Asymptotic</i>	<i>Path Length</i>	<i>N Max</i>	<i>Nstep</i>	<i>TU100</i>	<i>Max Dist</i>	<i>Lstep</i>	<i>Alt</i>	<i>ID</i>
31.78	31.61	264.37	0	0	20.000	-5.24	313.32	24.60006	0	65	0.00019	25.0005	0	20.0	195		
31.78	31.61	264.37	0	0	19.000	-7.71	314.82	24.65477	0	63	0.00023	25.0005	0	20.0	195		
31.78	31.61	264.37	0	0	18.000	-10.38	316.46	24.71653	0	65	0.00020	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	17.000	-13.23	318.28	24.78630	0	68	0.00017	25.0002	0	20.0	195		
31.78	31.61	264.37	0	0	16.000	-16.25	320.34	24.86599	0	71	0.00025	25.0005	0	20.0	195		
31.78	31.61	264.37	0	0	15.000	-19.38	322.70	24.95628	0	74	0.00021	25.0005	0	20.0	195		
31.78	31.61	264.37	0	0	14.000	-22.52	325.47	25.05902	0	77	0.00018	25.0005	0	20.0	195		
31.78	31.61	264.37	0	0	13.000	-25.48	328.74	25.17259	0	81	0.00023	25.0005	0	20.0	195		
31.78	31.61	264.37	0	0	12.000	-27.97	332.62	25.30893	0	86	0.00020	25.0002	0	20.0	195		
31.78	31.61	264.37	0	0	11.000	-29.57	337.09	25.46117	0	91	0.00024	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	10.000	-29.79	342.01	25.63626	0	98	0.00021	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	9.000	-28.23	347.18	25.84515	0	106	0.00024	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	8.000	-24.80	353.07	26.12190	0	115	0.00022	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	7.000	-19.57	2.30	26.58227	0	128	0.00027	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	6.000	-9.82	22.22	27.63240	0	146	0.00026	25.0003	0	20.0	195		
31.78	31.61	264.37	0	0	5.000	28.12	91.01	31.76201	0	184	0.00027	25.0001	0	20.0	195		
31.78	31.61	264.37	0	0	8.000	-24.80	353.07	26.12186	0	115	0.00027	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	7.900	-24.36	353.77	26.15621	0	116	0.00023	25.0002	0	20.0	195		
31.78	31.61	264.37	0	0	7.800	-23.91	354.50	26.19280	0	118	0.00023	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	7.700	-23.43	355.27	26.23120	0	118	0.00023	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	7.600	-22.94	356.09	26.27189	0	119	0.00023	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	7.500	-22.43	356.95	26.31518	0	123	0.00023	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	7.400	-21.90	357.88	26.36130	0	124	0.00023	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	7.300	-21.36	358.87	26.41059	0	126	0.00023	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	7.200	-20.79	359.93	26.46365	0	125	0.00023	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	7.100	-20.19	1.07	26.52062	0	125	0.00022	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	7.000	-19.57	2.30	26.58227	0	127	0.00022	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	6.900	-18.91	3.63	26.64900	0	129	0.00022	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	6.800	-18.22	5.07	26.72144	0	129	0.00022	25.0003	0	20.0	195		
31.78	31.61	264.37	0	0	6.700	-17.48	6.63	26.80064	0	132	0.00022	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	6.600	-16.69	8.33	26.88702	0	132	0.00023	25.0003	0	20.0	195		
31.78	31.61	264.37	0	0	6.500	-15.83	10.18	26.98206	0	134	0.00022	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	6.400	-14.89	12.19	27.08657	0	136	0.00022	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	6.300	-13.85	14.38	27.20215	0	138	0.00023	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	6.200	-12.68	16.76	27.33017	0	141	0.00023	25.0003	0	20.0	195		
31.78	31.61	264.37	0	0	6.100	-11.35	19.37	27.47286	0	143	0.00023	25.0004	0	20.0	195		
31.78	31.61	264.37	0	0	6.000	-9.82	22.22	27.63233	0	143	0.00023	25.0002	0	20.0	195		

	<i>Asymptotic</i>	<i>Path Length</i>	<i>N Max</i>	<i>Nstep</i>	<i>TU00</i>	<i>Max Dist</i>	<i>Lstep</i>	<i>Alt</i>	<i>ID</i>	<i>Lat</i>	<i>E Long</i>	<i>Re-Entrant</i>	
<i>GD Lat</i>	<i>GC Lat</i>	<i>E Long</i>	<i>Az</i>	<i>Rig</i>	<i>Path</i>	<i>Max</i>							
31.78	31.61	264.37	0	4.590	R	8.48565	2	4.83	0.00189	2.6048	0		
31.78	31.61	264.37	0	5.800	-5.96	28.75	0	148	0.00024	25.0004	0		
31.78	31.61	264.37	0	5.700	-3.50	32.51	0	151	0.00025	25.0004	0		
31.78	31.61	264.37	0	5.600	-0.56	36.69	0	155	0.00026	25.0004	0		
31.78	31.61	264.37	0	5.500	2.95	41.40	0	156	0.00028	25.0002	0		
31.78	31.61	264.37	0	5.400	7.14	46.84	0	160	0.00023	25.0004	0		
31.78	31.61	264.37	0	5.300	12.09	53.35	0	164	0.00024	25.0001	0		
31.78	31.61	264.37	0	5.200	17.79	61.65	0	170	0.00027	25.0003	0		
31.78	31.61	264.37	0	5.100	23.80	73.20	0	175	0.00022	25.0001	0		
31.78	31.61	264.37	0	5.000	28.12	91.01	0	185	0.00023	25.0003	0		
31.78	31.61	264.37	0	4.900	22.96	119.46	0	199	0.00025	25.0001	0		
31.78	31.61	264.37	0	4.800	-17.06	180.21	0	222	0.00024	25.0000	0		
31.78	31.61	264.37	0	4.700	1.62	276.29	0	372	0.00026	25.0004	0		
31.78	31.61	264.37	0	4.600	R	8.53329	2	444	0.00248	2.6167	0		
31.78	31.61	264.37	0	4.500	4.61	146.56	0	478	0.00025	25.0002	0		
31.78	31.61	264.37	0	4.400	12.08	149.30	0	365	0.00026	25.0001	0		
31.78	31.61	264.37	0	4.300	R	35.77379	9	1162	0.00099	3.0821	0		
31.78	31.61	264.37	0	4.200	-9.19	311.95	51.09002	5	770	0.00029	25.0004	0	
31.78	31.61	264.37	0	4.100	R	25.70248	4	713	0.00064	3.2452	0		
31.78	31.61	264.37	0	4.000	R	17.56007	3	709	0.00104	3.1831	0		
31.78	31.61	264.37	0	3.900	R	32.22328	7	1172	0.00067	3.0055	0		
31.78	31.61	264.37	0	3.800	R	27.97472	10	1584	0.00115	2.7353	0		
31.78	31.61	264.37	0	3.700	R	114.17108	43	5510	0.00082	2.8130	0		
31.78	31.61	264.37	0	3.600	R	8.25866	2	472	0.00068	2.2846	0		
31.78	31.61	264.37	0	3.500	R	29.13031	13	1935	0.00079	2.5803	0		
31.78	31.61	264.37	0	3.400	R	258.00621	101	12584	0.00075	3.3021	0		
31.78	31.61	264.37	0	3.300	-17.96	3523.41	379.80113	115	14200	0.00030	25.0003	0	
31.78	31.61	264.37	0	3.200	R	8.09547	3	592	0.00067	2.1716	0		
31.78	31.61	264.37	0	3.100	R	38.20262	21	2652	0.00067	2.4197	0		
31.78	31.61	264.37	0	3.000	2.95	41.40	28.81898	0	162	0.00025	25.0003	0	
31.78	31.61	264.37	0	2.900	3.33	41.91	28.85243	0	156	0.00023	25.0002	0	
31.78	31.61	264.37	0	2.800	3.73	42.42	28.88668	0	158	0.00024	25.0004	0	
31.78	31.61	264.37	0	2.700	R	4.13	42.94	0	158	0.0003	25.0003	0	
31.78	31.61	264.37	0	2.600	R	4.54	43.47	0	158	0.00024	25.0003	0	
31.78	31.61	264.37	0	2.500	4.95	44.01	28.89212	0	158	0.00025	25.0001	0	
31.78	31.61	264.37	0	2.400	5.37	44.56	29.02874	0	159	0.00025	25.0003	0	
31.78	31.61	264.37	0	2.300	5.80	45.11	29.06587	0	159	0.00025	25.0004	0	
31.78	31.61	264.37	0	2.200	R	6.24	45.68	0	158	0.00025	25.0001	0	
31.78	31.61	264.37	0	2.100	R	6.68	46.25	0	160	0.00025	25.0003	0	
31.78	31.61	264.37	0	2.000	R	7.14	46.84	0	160	0.00025	25.0002	0	

	<i>Asymptotic</i>	<i>Path Length</i>	<i>N Max</i>	<i>Nstep</i>	<i>TU00</i>	<i>Max Dist</i>	<i>Lstep</i>	<i>ID</i>
<i>GD Lat</i>	<i>GC Lat</i>	<i>E Long</i>	<i>Az</i>	<i>Rig</i>	<i>Path Length</i>	<i>N Max</i>	<i>Nstep</i>	<i>ID</i>
31.78	31.61	264.37	0	5.390	7.60	47.43	29.22081	0
31.78	31.61	264.37	0	5.380	8.07	48.04	29.26127	0
31.78	31.61	264.37	0	5.370	8.54	48.65	29.30260	0
31.78	31.61	264.37	0	5.360	9.02	49.28	29.34458	0
31.78	31.61	264.37	0	5.350	9.52	49.92	29.38749	0
31.78	31.61	264.37	0	5.340	10.02	50.58	29.43110	0
31.78	31.61	264.37	0	5.330	10.52	51.25	29.47567	0
31.78	31.61	264.37	0	5.320	11.04	51.93	29.52101	0
31.78	31.61	264.37	0	5.310	11.56	52.63	29.56727	0
31.78	31.61	264.37	0	5.300	12.09	53.35	29.61443	0
31.78	31.61	264.37	0	5.290	12.63	54.08	29.66279	0
31.78	31.61	264.37	0	5.280	13.18	54.83	29.71195	0
31.78	31.61	264.37	0	5.270	13.73	55.60	29.76226	0
31.78	31.61	264.37	0	5.260	14.29	56.39	29.81354	0
31.78	31.61	264.37	0	5.250	14.86	57.21	29.86613	0
31.78	31.61	264.37	0	5.240	15.44	58.04	29.91978	0
31.78	31.61	264.37	0	5.230	16.02	58.90	29.97454	0
31.78	31.61	264.37	0	5.220	16.60	59.79	30.03067	0
31.78	31.61	264.37	0	5.210	17.20	60.70	30.08829	0
31.78	31.61	264.37	0	5.200	17.79	61.65	30.14714	0
31.78	31.61	264.37	0	5.190	18.39	62.62	30.20732	0
31.78	31.61	264.37	0	5.180	19.00	63.63	30.26911	0
31.78	31.61	264.37	0	5.170	19.61	64.67	30.33362	0
31.78	31.61	264.37	0	5.160	20.21	65.75	30.39764	0
31.78	31.61	264.37	0	5.150	20.82	66.88	30.46442	0
31.78	31.61	264.37	0	5.140	21.43	68.04	30.53287	0
31.78	31.61	264.37	0	5.130	22.03	69.25	30.60365	0
31.78	31.61	264.37	0	5.120	22.63	70.52	30.67592	0
31.78	31.61	264.37	0	5.110	23.22	71.83	30.75086	0
31.78	31.61	264.37	0	5.100	23.80	73.20	30.82802	0
31.78	31.61	264.37	0	5.090	24.37	74.63	30.90736	0
31.78	31.61	264.37	0	5.080	24.92	76.13	30.98946	0
31.78	31.61	264.37	0	5.070	25.45	77.70	31.07393	0
31.78	31.61	264.37	0	5.060	25.95	79.34	31.16168	0
31.78	31.61	264.37	0	5.050	26.43	81.06	31.25235	0
31.78	31.61	264.37	0	5.040	26.87	82.86	31.34650	0
31.78	31.61	264.37	0	5.030	27.27	84.75	31.44440	0
31.78	31.61	264.37	0	5.020	27.61	86.74	31.54564	0
31.78	31.61	264.37	0	5.010	27.90	88.82	31.65173	0
31.78	31.61	264.37	0	5.000	28.12	91.01	31.76225	0

<i>Asymptotic Re-Entrant</i>													
<i>ID</i>	<i>Lat</i>	<i>Long</i>	<i>Alt</i>	<i>Step</i>	<i>Max Dist</i>	<i>Nstep</i>	<i>TU00</i>	<i>Path Length</i>	<i>NMax</i>	<i>Nstep</i>	<i>Max Dist</i>		
GD	<i>GC Lat</i>	<i>GC Long</i>	<i>Rig</i>	8.48565	2	483	0.00189	2.6048	0	20.0	195		
31.78	31.61	264.37	0	4.590	R	42.31149	2	755	0.00026	25.0004	0		
31.78	31.61	264.37	0	4.580	8.50	223.09	40.25146	2	657	0.00024	25.0003	0	
31.78	31.61	264.37	0	4.570	15.48	189.53	38.63423	3	621	0.00025	25.0001	0	
31.78	31.61	264.37	0	4.560	14.40	160.78	1127	0	25.0000	0	20.0	195	
31.78	31.61	264.37	0	4.550	-3.44	375.39	55.52940	7	458	0.00023	25.0003	0	
31.78	31.61	264.37	0	4.540	4.71	155.49	38.18150	2	444	0.00026	25.0003	0	
31.78	31.61	264.37	0	4.530	18.17	181.50	39.68981	2	531	0.00027	25.0001	0	
31.78	31.61	264.37	0	4.520	8.38	587.32	71.48544	9	1590	0.00027	25.0001	0	
31.78	31.61	264.37	0	4.510	12.49	226.39	42.24107	2	494	0.00027	25.0001	0	
31.78	31.61	264.37	0	4.500	4.61	146.56	37.55473	2	503	0.00027	25.0002	0	
31.78	31.61	264.37	0	4.490	15.27	156.54	38.16131	2	416	0.00026	25.0003	0	
31.78	31.61	264.37	0	4.480	4.04	359.25	52.04415	4	601	0.00024	25.0003	0	
31.78	31.61	264.37	0	4.470	R	30.63531	6	922	0.00076	2.9160	0	20.0	195
31.78	31.61	264.37	0	4.460	-4.38	195.20	41.55712	3	503	0.00027	25.0002	0	
31.78	31.61	264.37	0	4.450	-8.20	391.59	56.74052	7	834	0.00027	25.0001	0	
31.78	31.61	264.37	0	4.440	-17.87	271.53	45.37497	2	416	0.00026	25.0002	0	
31.78	31.61	264.37	0	4.430	8.38	166.72	38.55897	2	377	0.00025	25.0000	0	
31.78	31.61	264.37	0	4.420	4.88	145.24	37.28221	2	367	0.00023	25.0003	0	
31.78	31.61	264.37	0	4.410	7.96	140.45	37.01336	2	363	0.00025	25.0004	0	
31.78	31.61	264.37	0	4.400	12.08	149.30	37.52392	2	364	0.00029	25.0004	0	
31.78	31.61	264.37	0	4.390	8.32	179.00	39.31237	2	370	0.00025	25.0002	0	
31.78	31.61	264.37	0	4.380	14.63	39.09	56.62126	5	833	0.00025	25.0004	0	
31.78	31.61	264.37	0	4.370	2.01	653.07	76.55343	10	1474	0.00027	25.0004	0	
31.78	31.61	264.37	0	4.360	R	23.25382	6	1098	0.00080	2.4893	0	20.0	195
31.78	31.61	264.37	0	4.350	3.30	218.79	43.37173	4	527	0.00024	25.0004	0	
31.78	31.61	264.37	0	4.340	-19.57	329.38	51.29345	5	670	0.00024	25.0004	0	
31.78	31.61	264.37	0	4.330	-6.96	213.68	42.44818	3	456	0.00025	25.0001	0	
31.78	31.61	264.37	0	4.320	R	90.47529	31	3719	0.00060	3.0935	0	20.0	195
31.78	31.61	264.37	0	4.310	R	16.94065	4	705	0.00085	2.6886	0	20.0	195
31.78	31.61	264.37	0	4.300	R	35.76499	9	1180	0.00089	3.0920	0	20.0	195
31.78	31.61	264.37	0	4.290	-8.79	317.03	50.79638	5	816	0.00024	25.0004	0	
31.78	31.61	264.37	0	4.280	R	10.03041	2	417	0.00080	2.3634	0	20.0	195
31.78	31.61	264.37	0	4.270	R	9.98475	2	410	0.00096	2.3382	0	20.0	195
31.78	31.61	264.37	0	4.260	R	10.02752	2	394	0.00094	2.3751	0	20.0	195
31.78	31.61	264.37	0	4.250	R	10.14249	2	393	0.00079	2.4205	0	20.0	195
31.78	31.61	264.37	0	4.240	R	10.30465	2	396	0.00077	2.4666	0	20.0	195
31.78	31.61	264.37	0	4.230	R	10.50546	2	410	0.00071	2.5135	0	20.0	195
31.78	31.61	264.37	0	4.220	R	10.74947	2	432	0.00066	2.5609	0	20.0	195
31.78	31.61	264.37	0	4.210	R	11.42957	3	518	0.00098	2.6086	0	20.0	195
31.78	31.61	264.37	0	4.200	-9.19	317.95	51.08977	5	767	0.00025	25.0002	0	

Re-Entrant											
GD Lat	GC Lat	E Long	Az	Rig	Alt	Lat	E Long	ID	Alt	Lat	E Long
31.78	31.61	264.37	0	4.190	R	18.76410	4	758	0.00062	-32.8	487.2
31.78	31.61	264.37	0	4.180	R	24.31195	6	960	0.00065	195	44.7
31.78	31.61	264.37	0	4.170	442.61	59.07628	4	672	0.00025	20.0	195
31.78	31.61	264.37	0	4.160	5.98	534.71	9	1227	0.00026	20.0	195
31.78	31.61	264.37	0	4.150	22.85	791.83	90.01496	14	1643	0.00026	20.0
31.78	31.61	264.37	0	4.140	-4.83	653.54	75.37029	7	1277	0.00024	20.0
31.78	31.61	264.37	0	4.130	R	99.77461	29	3366	0.00096	3.0767	0
31.78	31.61	264.37	0	4.120	R	63.71266	16	2574	0.00066	3.0408	0
31.78	31.61	264.37	0	4.110	R	100.41292	18	2497	0.00072	3.3666	0
31.78	31.61	264.37	0	4.100	R	25.70266	4	739	0.00063	3.2452	0
31.78	31.61	264.37	0	4.090	-5.46	912.74	101.27636	15	2871	0.00027	25.0002
31.78	31.61	264.37	0	4.080	R	27.14181	4	717	0.00066	3.3689	0
31.78	31.61	264.37	0	4.070	R	27.02129	4	708	0.00066	3.3382	0
31.78	31.61	264.37	0	4.060	3.50	515.42	66.91958	7	1221	0.00028	25.0002
31.78	31.61	264.37	0	4.050	R	43.34165	9	1272	0.00073	3.2027	0
31.78	31.61	264.37	0	4.040	R	45.56245	12	1859	0.00071	3.2008	0
31.78	31.61	264.37	0	4.030	R	66.42011	18	2069	0.00065	3.2025	0
31.78	31.61	264.37	0	4.020	R	41.26292	10	1388	0.00080	3.1997	0
31.78	31.61	264.37	0	4.010	R	18.06813	3	682	0.00078	3.1929	0
31.78	31.61	264.37	0	4.000	R	17.56053	3	725	0.00109	3.1828	0
31.78	31.61	264.37	0	3.990	R	29.81871	8	1426	0.00378	3.1698	0
31.78	31.61	264.37	0	3.980	R	43.56670	10	1400	0.00067	3.1544	0
31.78	31.61	264.37	0	3.970	29.93	1170.28	124.76225	22	3323	0.00004	20.0
31.78	31.61	264.37	0	3.960	13.24	1222.95	127.15024	22	3286	0.00154	25.0004
31.78	31.61	264.37	0	3.950	3.76	832.00	93.47114	13	1887	0.00025	25.0001
31.78	31.61	264.37	0	3.940	R	31.31006	8	1350	0.00067	3.0746	0
31.78	31.61	264.37	0	3.930	R	16.15916	3	669	0.00075	3.0519	0
31.78	31.61	264.37	0	3.920	R	16.21055	3	581	0.00066	3.0283	0
31.78	31.61	264.37	0	3.910	R	106.79714	26	3662	0.00224	3.5057	0
31.78	31.61	264.37	0	3.900	R	32.222280	7	1151	0.00066	3.0056	0
31.78	31.61	264.37	0	3.890	R	54.16291	17	2101	0.00066	2.9590	0
31.78	31.61	264.37	0	3.880	R	41.26527	9	1214	0.00075	3.1118	0
31.78	31.61	264.37	0	3.870	R	28.11443	5	832	0.00068	3.1812	0
31.78	31.61	264.37	0	3.860	R	115.69314	34	4138	0.00089	3.1513	0
31.78	31.61	264.37	0	3.850	R	21.39112	4	808	0.00071	2.9571	0
31.78	31.61	264.37	0	3.840	R	42.52831	11	1454	0.00064	2.8326	0
31.78	31.61	264.37	0	3.830	R	15.15716	3	681	0.00075	2.8081	0
31.78	31.61	264.37	0	3.820	R	14.25701	3	598	0.00063	2.7838	0
31.78	31.61	264.37	0	3.810	R	38.77576	8	1344	0.00081	3.2256	0
31.78	31.61	264.37	0	3.800	R	27.97620	10	1605	0.00119	2.7355	0

Re-Entrant												
<i>GD Lat</i>	<i>GC Lat</i>	<i>E Long</i>	<i>Az</i>	<i>Rig</i>	<i>AltLat</i>	<i>AltLon</i>	<i>TU/00</i>	<i>Max Dist</i>	<i>Lat</i>	<i>E Long</i>	<i>ID</i>	
31.78	31.61	264.37	0	3.790	R	R	47.41895	15	1737	0.00069	2.9266	0
31.78	31.61	264.37	0	3.780	R	R	80.38117	26	3492	0.00079	3.0764	0
31.78	31.61	264.37	0	3.770	R	R	30.03156	8	1055	0.00065	2.9456	0
31.78	31.61	264.37	0	3.760	R	R	56.75616	19	2490	0.00083	2.9344	0
31.78	31.61	264.37	0	3.750	-6.45	635.61	81.76900	15	1843	0.00025	25.0004	0
31.78	31.61	264.37	0	3.740	R	R	23.11695	7	935	0.00071	2.8469	0
31.78	31.61	264.37	0	3.730	R	R	48.46938	17	1911	0.00089	2.6941	0
31.78	31.61	264.37	0	3.720	R	R	34.13973	11	1446	0.00289	2.9668	0
31.78	31.61	264.37	0	3.710	R	R	53.64331	19	2609	0.00068	2.8651	0
31.78	31.61	264.37	0	3.700	R	R	43.23410	14	2158	0.00077	2.8029	0
31.78	31.61	264.37	0	3.690	R	R	64.50743	24	3278	0.00089	2.8111	0
31.78	31.61	264.37	0	3.680	R	R	50.78450	22	3106	0.00085	2.6899	0
31.78	31.61	264.37	0	3.670	R	R	9.34822	4	651	0.00301	2.4374	0
31.78	31.61	264.37	0	3.660	R	R	8.93896	2	568	0.00064	2.4152	0
31.78	31.61	264.37	0	3.650	R	R	8.78802	2	547	0.00063	2.3933	0
31.78	31.61	264.37	0	3.640	R	R	8.65657	2	526	0.00069	2.3713	0
31.78	31.61	264.37	0	3.630	R	R	8.53741	2	506	0.00077	2.3496	0
31.78	31.61	264.37	0	3.620	R	R	8.43311	2	498	0.00076	2.3278	0
31.78	31.61	264.37	0	3.610	R	R	8.34060	2	478	0.00070	2.3061	0
31.78	31.61	264.37	0	3.600	R	R	8.23924	2	480	0.00070	2.2845	0
31.78	31.61	264.37	0	3.590	R	R	8.18702	2	477	0.00063	2.2630	0
31.78	31.61	264.37	0	3.580	R	R	8.12530	2	482	0.00063	2.2417	0
31.78	31.61	264.37	0	3.570	R	R	8.06972	2	484	0.00063	2.2204	0
31.78	31.61	264.37	0	3.560	R	R	8.02672	2	500	0.00063	2.2184	0
31.78	31.61	264.37	0	3.550	R	R	7.99231	2	532	0.00065	2.2180	0
31.78	31.61	264.37	0	3.540	R	R	7.97384	2	542	0.00073	2.2174	0
31.78	31.61	264.37	0	3.530	R	R	8.07911	3	591	0.00308	2.2167	0
31.78	31.61	264.37	0	3.520	R	R	8.17381	3	618	0.00083	2.2161	0
31.78	31.61	264.37	0	3.510	R	R	279.88047	117	13124	0.00077	3.2770	0
31.78	31.61	264.37	0	3.500	R	R	29.12923	13	1899	0.00082	2.5801	0
31.78	31.61	264.37	0	3.490	R	R	23.53714	10	1472	0.00069	2.3344	0
31.78	31.61	264.37	0	3.480	R	R	25.06072	11	1650	0.00084	2.4603	0
31.78	31.61	264.37	0	3.470	R	R	40.56292	19	2147	0.00070	2.5417	0
31.78	31.61	264.37	0	3.460	R	R	11.51278	5	738	0.00075	2.2117	0
31.78	31.61	264.37	0	3.450	R	R	45.37534	21	2220	0.00071	2.5701	0
31.78	31.61	264.37	0	3.440	R	R	36.79047	15	1831	0.00084	2.5995	0
31.78	31.61	264.37	0	3.430	R	R	14.86216	7	1000	0.00094	2.2567	0
31.78	31.61	264.37	0	3.420	R	R	67.22940	27	3053	0.00071	2.6238	0
31.78	31.61	264.37	0	3.410	R	R	10.84418	4	614	0.00077	2.2070	0
31.78	31.61	264.37	0	3.400	R	R	61.42813	31	3796	0.00064	2.5307	0

<i>Asymptotic</i>										<i>Re-Entrant</i>									
<i>GD Lat</i>	<i>GC Lat</i>	<i>E Long</i>	<i>Az</i>	<i>Ze</i>	<i>Rig</i>	<i>ALat</i>	<i>Alt</i>	<i>lon</i>	<i>Path Length</i>	<i>NMax</i>	<i>Nstep</i>	<i>TU100</i>	<i>Max Dist</i>	<i>Lstep</i>	<i>Alt</i>	<i>ID</i>	<i>Lat</i>	<i>E Long</i>	
31.78	31.61	264.37	0	0	3.390	R	R	R	40.84969	18	1897	0.00081	2.4727	0	20.0	195	30.9	633.6	
31.78	31.61	264.37	0	0	3.380	R	R	R	97.64354	45	4799	0.00151	2.5292	0	20.0	195	48.2	1133.3	
31.78	31.61	264.37	0	0	3.370	R	R	R	51.81226	23	2721	0.00072	2.5345	0	20.0	195	45.2	725.9	
31.78	31.61	264.37	0	0	3.360	R	R	R	40.58360	19	2217	0.00069	2.4465	0	20.0	195	31.9	627.9	
31.78	31.61	264.37	0	0	3.350	R	R	R	15.79704	6	840	0.00089	2.2999	0	20.0	195	-35.1	420.7	
31.78	31.61	264.37	0	0	3.350	R	R	R	29.61453	13	2002	0.00282	2.4926	0	20.0	195	49.3	527.3	
31.78	31.61	264.37	0	0	3.340	R	R	R	15.79598	6	846	0.00084	2.2961	0	20.0	195	-35.1	420.3	
31.78	31.61	264.37	0	0	3.330	R	R	R	41.25616	22	2328	0.00079	2.4769	0	20.0	195	-52.5	612.1	
31.78	31.61	264.37	0	0	3.320	R	R	R	31.71926	15	1881	0.00065	2.5071	0	20.0	195	-37.5	532.7	
31.78	31.61	264.37	0	0	3.310	R	R	R	89.32678	43	5059	0.00076	2.5618	0	20.0	195	-44.5	1061.4	
31.78	31.61	264.37	0	0	3.300	R	R	R	28.82309	14	2375	0.00263	2.4620	0	20.0	195	-34.8	506.9	
31.78	31.61	264.37	0	0	3.290	R	R	R	16.61049	7	1039	0.00066	2.4049	0	20.0	195	45.9	414.6	
31.78	31.61	264.37	0	0	3.280	R	R	R	54.69620	24	2507	0.00080	2.5099	0	20.0	195	-34.1	762.1	
31.78	31.61	264.37	0	0	3.270	R	R	R	20.45886	8	1101	0.00073	2.3813	0	20.0	195	47.9	446.9	
31.78	31.61	264.37	0	0	3.260	R	R	R	42.33894	21	2663	0.00111	2.4725	0	20.0	195	33.4	625.0	
31.78	31.61	264.37	0	0	3.250	R	R	R	17.17981	9	1368	0.00078	2.3088	0	20.0	195	47.8	409.9	
31.78	31.61	264.37	0	0	3.240	R	R	R	17.31153	10	1561	0.00078	2.2582	0	20.0	195	47.4	407.5	
31.78	31.61	264.37	0	0	3.230	R	R	R	8.50046	3	679	0.00067	2.2012	0	20.0	195	40.4	329.4	
31.78	31.61	264.37	0	0	3.220	R	R	R	8.24107	3	602	0.00060	2.1740	0	20.0	195	41.2	326.4	
31.78	31.61	264.37	0	0	3.210	R	R	R	8.09641	3	603	0.00069	2.1715	0	20.0	195	39.7	324.1	
31.78	31.61	264.37	0	0	3.200	R	R	R	8.13248	3	687	0.00165	2.1691	0	20.0	195	41.4	320.3	
31.78	31.61	264.37	0	0	3.190	R	R	R	16.75427	10	1569	0.00079	2.1665	0	20.0	195	46.6	398.4	
31.78	31.61	264.37	0	0	3.180	R	R	R	12.01466	7	917	0.00085	2.1641	0	20.0	195	-30.7	372.0	
31.78	31.61	264.37	0	0	3.170	R	R	R	11.61058	6	825	0.00086	2.1614	0	20.0	195	-31.6	369.3	
31.78	31.61	264.37	0	0	3.160	R	R	R	26.76304	16	2100	0.00145	2.2695	0	20.0	195	-33.0	478.7	
31.78	31.61	264.37	0	0	3.150	R	R	R	11.29537	6	785	0.00115	2.1557	0	20.0	195	-30.3	365.3	
31.78	31.61	264.37	0	0	3.140	R	R	R	33.33871	38	4517	0.00084	2.5447	0	20.0	195	50.8	871.1	
31.78	31.61	264.37	0	0	3.130	R	R	R	43.33471	20	2232	0.00136	2.4004	0	20.0	195	30.9	626.4	
31.78	31.61	264.37	0	0	3.120	R	R	R	58.17143	31	3151	0.00082	2.4348	0	20.0	195	-30.3	752.4	
31.78	31.61	264.37	0	0	3.110	R	R	R	42.32700	24	3059	0.00097	2.4197	0	20.0	195	-45.4	585.0	
31.78	31.61	264.37	0	0	3.100	R	R	R	33.85437	19	2487	0.00375	2.4165	0	20.0	195	-35.2	522.4	
31.78	31.61	264.37	0	0	3.090	R	R	R	26.50708	18	2140	0.00069	2.2242	0	20.0	195	-31.5	473.3	
31.78	31.61	264.37	0	0	3.080	R	R	R	8.39444	4	736	0.00062	2.1341	0	20.0	195	39.3	322.5	
31.78	31.61	264.37	0	0	3.070	R	R	R	8.12609	4	705	0.00065	2.1307	0	20.0	195	38.1	318.6	
31.78	31.61	264.37	0	0	3.060	R	R	R	8.31580	5	848	0.00112	2.1273	0	20.0	195	35.6	317.1	
31.78	31.61	264.37	0	0	3.050	R	R	R	16.18658	11	1452	0.00084	2.1238	0	20.0	195	46.5	387.4	
31.78	31.61	264.37	0	0	3.040	R	R	R	36.61931	23	2468	0.00294	2.3133	0	20.0	195	43.5	558.1	
31.78	31.61	264.37	0	0	3.030	R	R	R	37.89816	25	2662	0.00074	2.3556	0	20.0	195	43.7	566.1	
31.78	31.61	264.37	0	0	3.020	R	R	R	104.77082	54	5218	0.00078	2.4299	0	20.0	195	-34.2	1119.8	
31.78	31.61	264.37	0	0	3.010	R	R	R											

RUN END DATE 2000/12/23@22: 0:48
RUN START DATE 2000/12/23@22: 0:37
TOTAL NUMBER OF STEPS 311048.
TOTAL NUMBER OF TRAJECTORIES 316

Rigidity to Energy Conversion

Most people find rigidity a difficult concept to visualize. Rigidity is a canonical coordinate and the path of every particle having the same rigidity (independent of the atomic number or atomic charge) is identical. For this reason we have included a subroutine named AZRGEG that will perform rigidity to energy conversion (and vice-versa) for any element or isotope. The next two pages contain tables for rigidity to energy conversion. These tables are followed by the documentation for subroutine AZRGEG.

We have prepared two sample programs using this subroutine AZRGEG. The program ERG_RIG is an illustration of rigidity to energy rigidity conversion for a $^{16}\text{O}_8$ nuclei. Almost all elements (other than hydrogen) have a mass/charge ratio of approximately two. The program RIG_ERG is an illustration of energy to rigidity conversion for protons. We have entered the output files into a word processor and generated a one-page printout for each program. Then the two FORTRAN demonstration program listings follow.

¹⁶ O ₈ nuclei							
Rigidity (MV)	Energy (MeV/Nuc)	Rigidity (MV)	Energy (MeV/Nuc)	Rigidity (MV)	Energy (MeV/Nuc)	Rigidity (MV)	Energy (MeV/Nuc)
86.3	1.000	741.0	71.000	2199.9	510.000	7844.3	3100.000
122.1	2.000	746.4	72.000	2226.0	520.000	8049.7	3200.000
149.6	3.000	751.7	73.000	2252.0	530.000	8254.8	3300.000
172.8	4.000	757.1	74.000	2277.9	540.000	8459.7	3400.000
193.3	5.000	762.4	75.000	2303.7	550.000	8664.4	3500.000
211.8	6.000	767.6	76.000	2329.4	560.000	8868.9	3600.000
228.8	7.000	772.9	77.000	2354.9	570.000	9073.1	3700.000
244.6	8.000	778.1	78.000	2380.4	580.000	9277.2	3800.000
259.5	9.000	783.2	79.000	2405.7	590.000	9481.1	3900.000
273.7	10.000	788.4	80.000	2430.9	600.000	9684.9	4000.000
287.1	11.000	793.5	81.000	2456.1	610.000	9888.4	4100.000
299.9	12.000	798.6	82.000	2481.1	620.000	10091.9	4200.000
312.3	13.000	803.6	83.000	2506.1	630.000	10295.2	4300.000
324.1	14.000	808.7	84.000	2531.0	640.000	10498.4	4400.000
335.6	15.000	813.7	85.000	2555.8	650.000	10701.5	4500.000
346.7	16.000	818.7	86.000	2580.5	660.000	10904.4	4600.000
357.5	17.000	823.6	87.000	2605.1	670.000	11107.2	4700.000
367.9	18.000	828.6	88.000	2629.6	680.000	11310.0	4800.000
378.1	19.000	833.5	89.000	2654.1	690.000	11512.6	4900.000
388.0	20.000	838.3	90.000	2678.5	700.000	11715.2	5000.000
397.7	21.000	843.2	91.000	2702.8	710.000	11917.7	5100.000
407.2	22.000	848.0	92.000	2727.1	720.000	12120.0	5200.000
416.5	23.000	852.9	93.000	2751.3	730.000	12322.4	5300.000
425.5	24.000	857.6	94.000	2775.4	740.000	12524.6	5400.000
434.4	25.000	862.4	95.000	2799.4	750.000	12726.8	5500.000
443.1	26.000	867.2	96.000	2823.4	760.000	12928.8	5600.000
451.7	27.000	871.9	97.000	2847.4	770.000	13130.9	5700.000
460.1	28.000	876.6	98.000	2871.2	780.000	13332.9	5800.000
468.4	29.000	881.3	99.000	2895.0	790.000	13534.8	5900.000
476.5	30.000	886.0	100.000	2918.8	800.000	13736.6	6000.000
484.5	31.000	931.6	110.000	2942.5	810.000	13938.4	6100.000
492.4	32.000	975.4	120.000	2966.1	820.000	14140.2	6200.000
500.2	33.000	1017.8	130.000	2989.7	830.000	14341.9	6300.000
507.8	34.000	1058.9	140.000	3013.2	840.000	14543.5	6400.000
515.4	35.000	1098.8	150.000	3036.7	850.000	14745.1	6500.000
522.8	36.000	1137.7	160.000	3060.2	860.000	14946.7	6600.000
530.2	37.000	1175.6	170.000	3083.6	870.000	15148.2	6700.000
537.4	38.000	1212.6	180.000	3106.9	880.000	15349.7	6800.000
544.6	39.000	1248.9	190.000	3130.2	890.000	15551.2	6900.000
551.7	40.000	1284.5	200.000	3153.4	900.000	15752.6	7000.000
558.7	41.000	1319.4	210.000	3176.7	910.000	15954.0	7100.000
565.6	42.000	1353.7	220.000	3199.8	920.000	16155.3	7200.000
572.5	43.000	1387.4	230.000	3222.9	930.000	16356.6	7300.000
579.2	44.000	1420.6	240.000	3246.0	940.000	16557.9	7400.000
585.9	45.000	1453.4	250.000	3269.0	950.000	16759.1	7500.000
592.6	46.000	1485.7	260.000	3292.0	960.000	16960.3	7600.000
599.1	47.000	1517.5	270.000	3315.0	970.000	17161.5	7700.000
605.6	48.000	1549.0	280.000	3337.9	980.000	17362.7	7800.000
612.1	49.000	1580.1	290.000	3360.8	990.000	17563.8	7900.000
618.4	50.000	1610.8	300.000	3383.7	1000.000	17764.9	8000.000
624.7	51.000	1641.2	310.000	3610.3	1100.000	17966.0	8100.000
631.0	52.000	1671.3	320.000	3833.9	1200.000	18167.1	8200.000
637.2	53.000	1701.1	330.000	4055.1	1300.000	18368.1	8300.000
643.4	54.000	1730.6	340.000	4274.2	1400.000	18569.1	8400.000
649.5	55.000	1759.9	350.000	4491.5	1500.000	18770.1	8500.000
655.5	56.000	1788.9	360.000	4707.3	1600.000	18971.1	8600.000
661.5	57.000	1817.6	370.000	4921.7	1700.000	19172.0	8700.000
667.5	58.000	1846.1	380.000	5135.0	1800.000	19373.0	8800.000
673.4	59.000	1874.4	390.000	5347.3	1900.000	19573.9	8900.000
679.2	60.000	1902.5	400.000	5558.6	2000.000	19774.8	9000.000
685.0	61.000	1930.4	410.000	5769.2	2100.000	19975.7	9100.000
690.8	62.000	1958.1	420.000	5979.0	2200.000	20176.5	9200.000
696.5	63.000	1985.6	430.000	6188.1	2300.000	20377.4	9300.000
702.2	64.000	2013.0	440.000	6396.7	2400.000	20578.2	9400.000
707.9	65.000	2040.1	450.000	6604.8	2500.000	20779.0	9500.000
713.5	66.000	2067.1	460.000	6812.3	2600.000	20979.8	9600.000
719.1	67.000	2094.0	470.000	7019.4	2700.000	21180.6	9700.000
724.6	68.000	2120.7	480.000	7226.2	2800.000	21381.3	9800.000
730.1	69.000	2147.2	490.000	7432.5	2900.000	21582.1	9900.000
735.6	70.000	2173.6	500.000	7638.5	3000.000	21782.8	10000.000

PROTON		PROTON		PROTON		PROTON	
Rigidity (MV)	Energy (MeV)	Rigidity (MV)	Energy (MeV)	Rigidity (MV)	Energy (MeV)	Rigidity (MV)	Energy (MeV)
1.0	0.001	71.0	2.681	510.0	129.594	3200.0	2396.122
2.0	0.002	72.0	2.757	520.0	134.403	3300.0	2492.194
3.0	0.005	73.0	2.834	530.0	139.285	3400.0	2588.484
4.0	0.009	74.0	2.912	550.0	149.257	3500.0	2684.975
5.0	0.013	75.0	2.991	560.0	154.346	3600.0	2781.652
6.0	0.019	76.0	3.071	570.0	159.503	3700.0	2878.500
7.0	0.026	77.0	3.153	580.0	164.726	3800.0	2975.506
8.0	0.034	78.0	3.235	590.0	170.015	3900.0	3072.659
9.0	0.043	79.0	3.318	600.0	175.369	4000.0	3169.949
10.0	0.053	80.0	3.403	610.0	180.786	4100.0	3267.366
11.0	0.064	81.0	3.488	620.0	186.266	4200.0	3364.902
12.0	0.077	82.0	3.575	630.0	191.808	4300.0	3462.548
13.0	0.090	83.0	3.662	640.0	197.411	4400.0	3560.297
14.0	0.104	84.0	3.751	650.0	203.074	4500.0	3658.144
15.0	0.120	85.0	3.840	660.0	208.796	4600.0	3756.081
16.0	0.136	86.0	3.931	670.0	214.577	4700.0	3854.103
17.0	0.154	87.0	4.023	680.0	220.415	4800.0	3952.206
18.0	0.173	88.0	4.116	690.0	226.309	4900.0	4050.383
19.0	0.192	89.0	4.210	700.0	232.259	5000.0	4148.632
20.0	0.213	90.0	4.304	710.0	238.265	5100.0	4246.948
21.0	0.235	91.0	4.400	720.0	244.324	5200.0	4345.327
22.0	0.258	92.0	4.497	730.0	250.436	5300.0	4443.765
23.0	0.282	93.0	4.596	740.0	256.601	5400.0	4542.261
24.0	0.307	94.0	4.695	750.0	262.818	5500.0	4640.809
25.0	0.333	95.0	4.795	760.0	269.085	5600.0	4739.409
26.0	0.360	96.0	4.896	770.0	275.402	5700.0	4838.056
27.0	0.388	97.0	4.998	780.0	281.769	5800.0	4936.750
28.0	0.417	98.0	5.102	790.0	288.184	5900.0	5035.486
29.0	0.448	99.0	5.206	800.0	294.646	6000.0	5134.265
30.0	0.479	100.0	5.311	810.0	301.156	6100.0	5233.082
31.0	0.512	110.0	6.423	820.0	307.712	6200.0	5331.937
32.0	0.545	120.0	7.639	830.0	314.313	6300.0	5430.828
33.0	0.580	130.0	8.959	840.0	320.959	6400.0	5529.753
34.0	0.616	140.0	10.382	850.0	327.650	6500.0	5628.710
35.0	0.652	150.0	11.909	860.0	334.383	6600.0	5727.698
36.0	0.690	160.0	13.538	870.0	341.160	6700.0	5826.717
37.0	0.729	170.0	15.269	880.0	347.978	6800.0	5925.763
38.0	0.769	180.0	17.102	890.0	354.838	6900.0	6024.837
39.0	0.810	190.0	19.035	900.0	361.738	7000.0	6123.938
40.0	0.852	200.0	21.069	910.0	368.679	7100.0	6223.063
41.0	0.895	210.0	23.203	920.0	375.659	7200.0	6322.212
42.0	0.939	220.0	25.435	930.0	382.677	7300.0	6421.384
43.0	0.984	230.0	27.766	940.0	389.734	7400.0	6520.578
44.0	1.031	240.0	30.194	950.0	396.829	7500.0	6619.793
45.0	1.078	250.0	32.720	960.0	403.961	7600.0	6719.029
46.0	1.126	260.0	35.341	970.0	411.128	7700.0	6818.285
47.0	1.176	270.0	38.058	980.0	418.332	7800.0	6917.559
48.0	1.226	280.0	40.869	990.0	425.571	7900.0	7016.852
49.0	1.278	290.0	43.774	1000.0	432.845	8000.0	7116.162
50.0	1.331	300.0	46.772	1100.0	507.375	8100.0	7215.489
51.0	1.384	310.0	49.862	1200.0	584.825	8200.0	7314.832
52.0	1.439	320.0	53.043	1300.0	664.773	8300.0	7414.191
53.0	1.495	330.0	56.315	1400.0	746.862	8400.0	7513.565
54.0	1.552	340.0	59.676	1500.0	830.796	8500.0	7612.953
55.0	1.610	350.0	63.126	1600.0	916.323	8600.0	7712.356
56.0	1.669	360.0	66.663	1700.0	1003.234	8700.0	7811.772
57.0	1.729	370.0	70.287	1800.0	1091.350	8800.0	7911.202
58.0	1.790	380.0	73.996	1900.0	1180.521	8900.0	8010.644
59.0	1.852	390.0	77.791	2000.0	1270.620	9000.0	8110.098
60.0	1.916	400.0	81.669	2100.0	1361.537	9100.0	8209.565
61.0	1.980	410.0	85.631	2200.0	1453.179	9200.0	8309.042
62.0	2.045	420.0	89.674	2300.0	1545.466	9300.0	8408.531
63.0	2.112	430.0	93.798	2400.0	1638.328	9400.0	8508.031
64.0	2.179	440.0	98.003	2500.0	1731.706	9500.0	8607.541
65.0	2.248	450.0	102.286	2600.0	1825.548	9600.0	8707.062
66.0	2.317	460.0	106.648	2700.0	1919.807	9700.0	8806.592
67.0	2.388	470.0	111.087	2800.0	2014.443	9800.0	8906.132
68.0	2.460	480.0	115.602	2900.0	2109.423	9900.0	9005.680
69.0	2.532	490.0	120.192	3000.0	2204.713	10000.0	9105.238
70.0	2.606	500.0	124.856	3100.0	2300.288		

Subroutine AZRGEG (NA, NZ, PAMU, RIGIN, EPN, BETA)

This is a subroutine for rigidity to energy conversion and vice-versa

Input arguments:

NA	Integer atomic number
NZ	Integer atomic charge
PAMU	Physical mass unit of element (or isotope)

Utility arguments

RIGIN	Rigidity in MV
EPN	Energy per nucleon in MeV

Return Arguments

BETA	Particle speed as a fraction of light speed (v/c)
------	---

Labeled Common arguments: none**Dimensioned Variables** none**Data files:** none**Operation:**

When the subroutine is called it can do either a rigidity-to-energy conversion or an energy-to-rigidity conversion. If the variable RIGIN is positive, then the conversion is rigidity-to-energy and the output energy is in the variable EPN. If the variable RIGIN is initially set to zero or a negative value, then the conversion is energy-to-rigidity. The value RIGIN is replaced with the appropriate rigidity value for the energy specified in the variable EPN. The subroutine will work for any element or isotope. It is necessary to specify the atomic number, the atomic charge and the atomic mass in physical mass units. If these values are not specifically provided, the program will default to the values appropriate for protons. The variable RIGIN must be in the rigidity unit MV. The variable EPN (energy per nucleon) will be in MeV.

When called with the proper arguments, the software first tests the value of the variable RIGIN to determine which conversion to perform. The total kinetic energy of the particle is computed. The next step is to compute the relativistic gamma. (The relativistic gamma factor is a "natural unit" used in high-energy physics that is the ratio of the particle kinetic energy to rest mass energy.) From the relativistic gamma factor, either energy or rigidity can be computed. The particle speed BETA can also be computed from the relativistic gamma factor. The particle speed BETA is returned as an argument since it is useful in many conversions such as differential flux in terms of energy or rigidity.

```
program ERG_RIG

c.....+.....+.....+.....+.....+.....+.....+...
c      Energy to Rigidity conversion example
c.....+.....+.....+.....+.....+.....+...
c      \/ Example of the use of the Energy to Rigidity conversion program
c          Convert geomagnetic cutoff rigidity to proton cutoff energy
c          need to define proton atomic number, charge and mass
c          rigidity is often used in units of GV;
c          subroutine azrgeg needs rigidity in units of MV
c          subroutine azrgeg returns energy in units of Mev per nucleon
c.....+.....+.....+.....+.....+.....+...
c      Programmed by Don F. Smart (sssrc@msn.com)
c      Note - programming adheres to convention that variables beginning
c              with i, j, k, l, m, n are integer values,
c              variables beginning with c are character variables
c              all other variables are real*8
c.....+.....+.....+.....+.....+...
c
c      implicit integer (i-n)
c      implicit REAL*8 (a-b)
c      implicit REAL*8 (d-h)
c      implicit REAL*8 (o-z)

c.....+.....+.....+.....+.....+...
c
c      open (7,file='erg-rig.txt', status='unknown')

c.....+.....+.....+.....+.....+...
c      Define atomic number, atomic charge and rest mass for oxygen16
c      (Note, any element or isotope can be specified)
c.....+.....+.....+.....+...
c
c      na = 16
c      nz = 8
c      pamu = 16.00

c.....+.....+.....+.....+.....+...
c      \/ First demonstration does Energy to Rigidity from 1 to 100 MeV
c
c      Do 100 i=1,100
c          rpmv = 0.0
c          epnmev = float (i)
c          call azrgeg (na,nz,pamu,rpmv,epnmev,beta)

c
c      write (7, 1070) rpmv,epnmev

100 continue

c.....+.....+.....+.....+.....+...
c      \/ Second demonstration does Energy to Rigidity
c          from 100 to 1000 MeV in 10 MeV increments
c.....+.....+.....+.....+...
```

```
Do 110 i=100,1000,10
  rpmv = 0.0
  epnmev = float (i)
  call azrgeg (na,nz,pamu,rpmv,epnmev,beta)

  write (7, 1070)  rpmv,epnmev
110 continue
c.....+.....+.....+.....+.....+.....+.....+...
c     \/ Second demonstration does Energy to Rigidity
c           from 1000 to 10000 MeV in 100 MeV increments
c.....+.....+.....+.....+.....+.....+...
c.....+.....+.....+.....+.....+.....+...
c
  Do 120 i=1000,10000,100
    rpmv = 0.0
    epnmev = float (i)
    call azrgeg (na,nz,pamu,rpmv,epnmev,beta)

    write (7, 1070)  rpmv,epnmev
120 continue

1070 format (1f10.1, 1f10.3)

      stop
      end
      subroutine azrgeg (na,nz,pamu,rigin,epn,beta)
c.....+.....+.....+.....+.....+.....+...
c     subroutine to convert rigidity to energy and visa versa
cLast mod 17 March 94
c.....+.....+.....+.....+.....+.....+...
c     Programmed by Don F. Smart (sssrc@msn.com)
c     Note - programming adheres to convention that variables beginning
c           with i, j, k, l, m, n   are integer values,
c           variables beginning with c are character variables
c           all other variables are real*4
c.....+.....+.....+.....+.....+.....+...
c
      implicit integer (i-n)
      implicit REAL*8 (a-b)
      implicit REAL*8 (d-h)
      implicit REAL*8 (o-z)
c.....+.....+.....+.....+.....+.....+...
c
c     write (*,6010) na,nz,pamu,rigin,epn          ! diag
c6010 format (' azrgeg1 ',2i5,3f15.3)          ! diag
c
c.....+.....+.....+.....+.....+.....+...
c     Check, if na, nz, or pamu not specified, put in default for protons
c           epamu is rest mass energy per atomic mass unit
c.....+.....+.....+.....+.....+...
c
      if (pamu.le.0.0)  pamu = 1.0081451
      if (na.le.0)      na = 1
      if (nz.le.0)      nz = 1
c
```

```

epamu = 931.141
c
anuc = na
zcharg = nz
rsmspn = (pamu/anuc)*epamu
c
trig = rigin
c.....+.....+.....+.....+.....+.....+...
c     \/ if trig .le. 0.0      do energy   to rigidity conversion
c         if trig .gt. 0.0      do rigidity to energy conversion
c.....+.....+.....+.....+.....+...
c
if (trig.le.0.0)  then                                Energy to Rigidity conversion
c
gmaeg = (anuc*epn+anuc*rsmspn)/(anuc*rsmspn)
gmaegg = (epn+rsmspn)/rsmspn
rigin = dsqrt(gmaeg*gmaeg-1.0)*rsmspn*anuc/zcharg
relgma = gmaeg
else
Rigidity to Energy conversion
c
gmarg = dsqrt(((rigin*zcharg)/(rsmspn*anuc))**2+1.0)
epn = (gmarg-1.0)*rsmspn
relgma = gmarg
endif
c
write (*,6020) anuc,zcharg,rsmspn,rigin,epn,gmaeg,gmaegg,gmarg      ! diag
c
write (*,6030) na,nz,pamu,rigin,epn                                  ! diag
c6020 format (' azrgeg2 ',8f10.3)                                         ! diag
c6030 format (' azrgeg3 ',2i5,3f15.5)                                       ! diag
c
beta = dsqrt(1.0-1.0/(relgma*relgma))
c
return
c
beta is v/c (speed as fraction of light speed)
c
energy in MeV
c
epamu is mass-energy conversion = 931.141 MeV per amu
c
epn is kinetic energy per nucleon
c
na is atomic number
c
nz is charge
c
pamu is rest mass in physical atomic mass units
c
relgam is relativistic factor 'gamma'
c
rigidity in mv
c
rsmspn is rest mass per nucleon in MeV
c
end

```

```
program RIG_ERG
c.....+.....+.....+.....+.....+.....+.....+...
c      Rigidity to energy conversion example
c.....+.....+.....+.....+.....+.....+...
c      \/ Example of the use of the Rigidity to Energy conversion program
c          Convert geomagnetic cutoff rigidity to proton cutoff energy
c              need to define proton atomic number, charge and mass
c                  rigidity is often used in units of GV;
c          Subroutine azrgeg needs rigidity in units of MV
c          Subroutine azrgeg returns energy in units of Mev per nucleon
c.....+.....+.....+.....+.....+.....+...
c      Programmed by Don F. Smart (sssrc@msn.com)
c      Note - programming adheres to convention that variables beginning
c          with i, j, k, l, m, n are integer values,
c          variables beginning with c are character variables
c          all other variables are real*8
c.....+.....+.....+.....+.....+...
c
c      implicit integer (i-n)
c      implicit REAL*8 (a-b)
c      implicit REAL*8 (d-h)
c      implicit REAL*8 (o-z)
c.....+.....+.....+.....+.....+...
c
c      open (7,file='rig-err.txt', status='unknown')
c.....+.....+.....+.....+.....+...
c
c      Define atomic number, atomic charge and rest mass for proton
c.....+.....+.....+.....+...
c
c      na = 1
c      nz = 1
c      pamu = 1.0081451
c.....+.....+.....+.....+.....+...
c
c      \/ First demonstration does Rigidity to Energy from 1 to 100 MV
c.....+.....+.....+.....+...
c
c      Do 100 i=1,100
c          rpmv = float (i)
c          call azrgeg (na,nz,pamu,rpmv,epnmev,beta)           !sub20
c
c          write (7, 1070) rpmv,epnmev
c
c      100 continue
c.....+.....+.....+.....+.....+...
c
c      \/ Second demonstration does Rigidity to Energy
c          from 100 to 1000 MV in 10 MV increments
c.....+.....+.....+.....+...
c
c      Do 110 i=100,1000,10
c          rpmv = float (i)
```

```

        call azrgeg (na,nz,pamu,rpmv,epnmev,beta)

        write (7, 1070) rpmv,epnmev
110 continue
c
c.....+.....+.....+.....+.....+.....+.....+...
c   \/ Third demonstration does Rigidity to Energy
c           from 1000 to 10000 MV in 100 MV increments
c.....+.....+.....+.....+.....+.....+.....+...
c
c   Do 120 i=1000,10000,100
      rpmv = float (i)
      call azrgeg (na,nz,pamu,rpmv,epnmev,beta)

      write (7, 1070) rpmv,epnmev
120 continue

1070 format (1f10.1, 1f10.3)

      stop
      end
      subroutine azrgeg (na,nz,pamu,rigin,epn,beta)
c
c.....+.....+.....+.....+.....+.....+.....+...
c   subroutine to convert rigidity to energy and visa versa
cLast mod 17 March 94
c.....+.....+.....+.....+.....+.....+.....+...
c   Programmed by Don F. Smart (sssrc@msn.com)
c   Note - programming adheres to convention that variables beginning
c           with i, j, k, l, m, n are integer values,
c           variables beginning with c are character variables
c           all other variables are real*4
c.....+.....+.....+.....+.....+.....+...
c
c   implicit integer (i-n)
c   implicit REAL*8 (a-b)
c   implicit REAL*8 (d-h)
c   implicit REAL*8 (o-z)
c
c.....+.....+.....+.....+.....+.....+...
c
c   write (*,6010) na,nz,pamu,rigin,epn                      ! diag
c6010 format (' azrgegl ',2i5,3f15.3)                         ! diag
c
c.....+.....+.....+.....+.....+.....+.....+...
c   Check, if na, nz, or pamu not specified, put in default for protons
c           epamu is rest mass energy per atomic mass unit
c.....+.....+.....+.....+.....+.....+...
c
c   if (pamu.le.0.0) pamu = 1.0081451
c   if (na.le.0)      na = 1
c   if (nz.le.0)      nz = 1
c
c   epamu = 931.141
c
c   anuc = na
c   zcharg = nz

```

```

        rsmspn = (pamu/anuc)*epamu
c
c      trig = rigin
c
c.....+.....+.....+.....+.....+.....+...
c  \/ if trig .le. 0.0    do energy   to rigidity conversion
c      if trig .gt. 0.0    do rigidity to energy conversion
c.....+.....+.....+.....+.....+.....+...
c
c      if (trig.le.0.0)  then          Energy to Rigidity conversion
c
c      gmaeg = (anuc*epn+anuc*rsmspn) / (anuc*rsmspn)
c      gmaegg = (epn+rsmspn)/rsmspn
c      rigin = dsqrt(gmaeg*gmaeg-1.0)*rsmspn*anuc/zcharg
c      relgma = gmaeg
c
c      else                      R rigidity to Energy conversion
c
c      gmarg = dsqrt(((rigin*zcharg)/(rsmspn*anuc))**2+1.0)
c      epn = (gmarg-1.0)*rsmspn
c      relgma = gmarg
c      endif
c
c      write (*,6020) anuc,zcharg,rsmspn,rigin,epn,gmaeg,gmaegg,gmarg      ! diag
c      write (*,6030) na,nz,pamu,rigin,epn                                ! diag
c6020 format (' azrgeg2 ',8f10.3)                                         ! diag
c6030 format (' azrgeg3 ',2i5,3f15.5)                                         ! diag
c
c      beta = dsqrt(1.0-1.0/(relgma*relgma))
c
c      return
c
c      beta is v/c (speed as fraction of light speed)
c      energy in MeV
c      epamu is mass-energy conversion = 931.141 MeV per amu
c      epn is kinetic energy per nucleon
c      na is atomic number
c      nz is charge
c      pamu is rest mass in physical atomic mass units
c      relgam is relativistic factor 'gamma'
c      rigidity in mv
c      rsmspn is rest mass per nucleon in MeV
c
c      end

```

```
1      PROGRAM TJI95
2      C
3      C.....+.....+.....+.....+.....+.....+.....+...
4      C      Multi-platform COSMIC-RAY TRAJECTORY PROGRAM
5      C      FORTRAN 77   transportable version
6      C      Read in control card; LAT, LON, RIG, ZENITH, AZIMUTH, DELPC, INDO
7      C          Then calculate      INDO      trajectories
8      C          Starting at      PC
9      C          Incrementing at    DELPC    intervals
10     C      Includes conversion from Geodetic to Geocentric coordinates
11     C      Includes re-entrant albedo calculations
12     C      Uses subroutine SINGLTJE to do trajectory calculations
13     C      Magnetic field - IGRF 1995 (order 10)      #####
14     C.....+.....+.....+.....+.....+.....+.....+...
15     C      Restrictions: Cannot run over N or S pole; will get BETA blowup
16     C.....+.....+.....+.....+.....+.....+...
17     C      Mod History
18     CLast Mod 21 Dec 00  Make all intrinsic function double precision for PC
19     C      Mod 20 Dec 00  Insert 8 character format 1000 with AZ & ZE
20     C      Mod 17 Feb 99  set limit to 600000
21     C      Mod 17 Feb 99  if (ymax.lt.6.6) IFATE = 3
22     C      Mod  Aug 97  Adjust step size to minimize beta problems
23     C      Mod  Jan 97  High latitude step size adjust, introduce AHLT
24     C      Mod  Jun 96  EDIF limit set to 1.0e-5
25     C      Mod  Jun 96  IERRPT formats, Boundary and look ahead
26     C      Mod  Feb 96  Standard reference TJ1V line check
27     C      Mod  Dec 94  Print out start and end times of PC run
28     C*****+
29     C      Timing estimates base on COMPAQ Digital FORTRAN
30     C      Will run on PIII PC at 850 MHZ      55000 steps/sec (Real*8)
31     C      Will run on PIII PC at 700 MHZ      39000 steps/sec (Real*8)
32     C      Will run on PIII PC at 550 MHZ      32000 steps/sec (Real*8)
33     C      Will run on PII PC at 400 MHZ      23000 steps/sec (Real*8)
34     C*****+
35     C      * TAPE*      Monitor program operation
36     C      * TAPE1      Trajectory control cards
37     C      * TAPE7      80 character line (card image) output
38     C      * TAPE8      132 character line printer output
39     C      * TAPE16     Diagnostic output for trouble shooting
40     C      *          Normally turned off (open statement commented out)
41     C*****+
42     C.....+.....+.....+.....+.....+...
43     C      Programmer - Don F. Smart; FORTRAN77
44     C      Note - The programming adheres to the conventional FORTRAN
45     C          default standard that variables beginning with
46     C          'i','j','k','l','m',or 'n' are integer variables
47     C          Variables beginning with "c" are character variables
48     C          All other variables are real
49     C.....+.....+.....+.....+.....+...
50     C      Do not mix different type variables in same common block
51     C      Some computers do not allow this
52     C.....+.....+.....+.....+...
53     C
54     IMPLICIT INTEGER (I-N)
55     IMPLICIT REAL * 8(A-B)
56     IMPLICIT REAL * 8(D-H)
57     IMPLICIT REAL * 8(O-Z)
58     C
59     C.....+.....+.....+.....+...
60     C
61     COMMON /WRKVLU/ F(6),Y(6),ERAD,EOMC,VEL,BR,BT,BP,B
62     COMMON /WRKTSC/ TSY2,TCY2,TSY3,TCY3
63     COMMON /TRIG/ PI,RAD,PIO2
64     COMMON /GEOID/ ERADPL, ERECSQ
65     COMMON /SNGLR/ SALT,DISOUT,GCLATD,GDLATD,GLOND,GDAZD,GDZED,
66     *                  RY1,RY2,RY3,RHT,TSTEP
67     COMMON /SNGLI/ LIMIT,NTRAJC,IERRPT
68     C
69     C.....+.....+.....+.....+...
70     C
```

```

71      OPEN (1, FILE='TAPE1', STATUS='OLD')
72      OPEN (7, FILE='TAPE7', STATUS='UNKNOWN')
73      OPEN (8, FILE='TAPE8', STATUS='UNKNOWN')
74      open (16,FILE='TAPE16',STATUS='UNKNOWN')
75      C
76      C.....+.....+.....+.....+.....+.....+.....+...
77      C  \/ User defined program control
78      C.....+.....+.....+.....+.....+.....+...
79
80      FSTEP = 4.0E08
81      LIMIT = 600000
82      C
83      C.....+.....+.....+.....+.....+.....+...
84      C  \/ FSTEP is total number of steps before run is terminated
85      C      LIMIT is max number of steps before trajectory declared F
86      C.....+.....+.....+.....+.....+.....+...
87      C  \/ Define program constants
88      C.....+.....+.....+.....+.....+...
89      C      DISOUT is radial distance for trajectory termination
90      C      ERAD is average earth radius
91      C      NTRAJC is number of trajectory computed in this run
92      C      RHT is top of atmosphere for re-entrant trajectory
93      C      TSTEP is number of steps executed in this run
94      C.....+.....+.....+.....+...
95      C
96      NTRAJC = 0
97      TSTEP = 0.0
98      C
99      DISOUT = 25.0
100     ERAD = 6371.2
101     RHT = 20.0
102     VEL = 2.99792458E5/ERAD
103     C
104     C.....+.....+.....+.....+...
105     C      "VEL" is light velocity in earth radii per second
106     C      Light speed defined as 299792458 m/s
107     C      Ref: E. R. Cohen AND B. N. Taylor, "The Fundamental Physical
108     C      Constants, Physics Today P11, August 1987.
109     C.....+.....+.....+...
110     C  \/ Define essential trigonometric values
111     C.....+.....+.....+...
112     C
113     PI = ACOS(-1.0)
114     RAD = 180.0/PI
115     PIO2 = PI/2.0
116     C
117     C.....+.....+.....+...
118     C  \/ TAPE1 must contain trajectory control cards
119     C      Terminate program if no data on TAPE1 file
120     C      Terminate if EOF encountered
121     C      Terminate if negative data found on input file
122     C      Terminate if bad data found on input file
123     C.....+.....+.....+...
124     C
125     100 READ (1,1010,IOSTAT=IOSTAT,ERR=120,END=110) GDLATD,GLOND,PC,
126     *      GDZED,GDAZD,DELPC,INDO,IERRPT,INDEX
127     1010 FORMAT (BZ,6F8.2,3I8)
128     C
129     110 CONTINUE
130     IF (IOSTAT.LT.0) THEN
131         WRITE (*,1020)
132         GO TO 150
133     ENDIF
134     1020 FORMAT (' END OF FILE ON TAPE 1 (DATA INPUT)')
135
136     120 IF (IOSTAT.GT.0) THEN
137         WRITE (*,1030) IOSTAT,GDLATD,GLOND,PC,DELPC,
138         *      INDO,IERRPT,INDEX
139         GO TO 150
140     ENDIF

```

```

141 1030 FORMAT (' ERROR ON DATA INPUT FILE (TAPE1), IOSTAT =',I5/
142      * 4F8.3,3I8)
143 C
144     IF (PC.LE.0) THEN
145         WRITE (*,1040)
146         GO TO 150
147     ENDIF
148 1040 FORMAT (' END OF DATA INPUT (NEGATIVE VALUE READ IN)')
149 C
150     WRITE (*,1050) GDLATD,GLOND,PC,GDZED,GDAZD,DELPC,INDO,IERRPT,INDEX
151 1050 FORMAT (' TAPE 1 ',6F7.2,3I6)
152 C
153 C.....+.....+.....+.....+.....+.....+...
154 C     \/ Start at top of atmosphere (20 km above surface of oblate earth)
155 C     Coding is relic of past when ISALT was read in
156 C.....+.....+.....+.....+.....+...
157 C
158     ISALT = 0
159     IF (ISALT.LE.0) SALT = 20.0
160     IF (ISALT.GT.0) SALT = ISALT
161 C
162     KNT = 0
163     IDELPC = DELPC*1000.0+0.0001
164     INDXPC = PC*1000.0+0.0001
165 C
166 C.....+.....+.....+.....+.....+...
167 C     For trajectories from Earth
168 C         convert from Geodetic coordinates to Geocentric coordinates
169 C             Geodetic coordinates used for input
170 C             Geocentric coordinates used for output
171 C         All calculation are done in Geocentric coordinates!
172 C     \/ Conversion from Geodetic to Geocentric coordinates
173 C.....+.....+.....+.....+...
174 C
175     CALL GDGC (TCD, TSD)
176 C
177 C.....+.....+.....+.....+...
178 C     \/ Remember position of initial point on trajectory
179 C         in Geocentric coordinates
180 C         Y(1) is distance in earth radii from geocenter
181 C             Start with height above geoid and convert to earth radii
182 C             The initial values of Y(1), Y(2), and Y(3) are
183 C                 calculated in subroutine GDGC
184 C         Coordinate reference system
185 C             Y(1) = R      = vertical
186 C             Y(2) = THETA = south
187 C             Y(3) = PHI   = east
188 C.....+.....+.....+...
189 C
190     RY2 = Y(2)
191     RY3 = Y(3)
192     RY1 = Y(1)
193 C
194     GDAZ = GDAZD/RAD
195     GDZE = GDZED/RAD
196     TSGDZE = SIN(GDZE)
197     TCGDZE = COS(GDZE)
198     TSGDAZ = SIN(GDAZ)
199     TCGDAZ = COS(GDAZ)
200 C
201 C.....+.....+.....+...
202 C     \/ Get Y1, Y2, Y3 components in Geodetic coordinates
203 C         Azimuth is measured clockwise from the north
204 C             in R, THETA, PHI coordinates, in the THETA-PHI plane
205 C             The angle is 180 - AZD
206 C.....+.....+...
207 C
208     Y1GD = TCGDZE
209     Y2GD = -TSGDZE*TCGDAZ
210     Y3GD = TSGDZE*TSGDAZ

```

```

211 C
212 C.....+.....+.....+.....+.....+.....+..+
213 C   \/ The small angle delta at the point in space between the
214 C     downward Geodetic direction and the
215 C     downward Geocentric direction is given by
216 C     DELTA = Geocentric co-latitude + Geodetic latitude - 90 (deg)
217 C
218 C     We are looking up
219 C       The rotation from Geodetic vertical to Geocentric Vertical
220 C         Is always rotation toward the equator
221 C
222 C   \/ Convert from Geodetic to Geocentric Components for Y1, Y2,
223 C.....+.....+.....+.....+.....+..+
224 C
225 Y1GC = Y1GD*TCD+Y2GD*TSD
226 Y2GC = -Y1GD*TSD+Y2GD*TCD
227 Y3GC = Y3GD
228 C
229 C   WRITE (*,1060) GDZED,GDZE,GDAZD,GDAZ,TSGDZE,TCGDZE,TSGDAZ,TCGDAZ
230 C   WRITE (*,1060) Y1GD,Y2GD,Y3GD,Y1GC,Y2GC,Y3GC
231 C1060 FORMAT (' 1050',8F15.5)
232 C
233 C.....+.....+.....+.....+.....+.....+..+
234 C   ****
235 C   Main control of trajectory calculations begins here
236 C   Trajectories are calculated in subroutine SINGLTJ
237 C   ****
238 C
239 C   PC      = rigidity IN GV
240 C   INDXPC = index of rigidity in MV (integer)
241 C   IRSLT  = trajectory result
242 C     IRSLT    +1    allowed
243 C     IRSLT    0    failed
244 C     IRSLT   -1   re-entrant
245 C.....+.....+.....+.....+.....+..+
246 C
247 DO 130 NDO = 1, INDO
248 C
249 IF (IERRPT.GE.1) WRITE (16,1070) GDLATD,GLOND,KNT,INDO,NDO,
250 *                   IDELPC,INDXPC,DELPC, PC
251 C
252 CALL SINGLTJ (PC,IRSLT,INDXPC,Y1GC,Y2GC,Y3GC)
253 C
254 KNT = KNT+1
255 INDXPC = INDXPC-IDELPC
256 PC = FLOAT(INDXPC)/1000.0
257 C
258 +.....+.....+.....+.....+.....+..+
259 C   \/ Check termination conditions
260 C +.....+.....+.....+.....+.....+..+
261 C
262 IF (PC .LE. 0.0) GO TO 140
263 IF (TSTEP .GE. FSTEP) GO TO 150
264 C
265 130 CONTINUE
266 140 CONTINUE
267 1070 FORMAT (' 1070 ',2F7.2,5I6,2F6.2)
268 C
269 C.....+.....+.....+.....+.....+..+
270 C   ****
271 C   End of main control loop
272 C   ****
273 C   /\ Go read in next control card
274 C.....+.....+.....+.....+.....+..+
275 C
276 GO TO 100
277 C
278 C.....+.....+.....+.....+.....+..+
279 C   ****
280 C   End of trajectory calculations

```

```

281 C ****
282 C.....+.....+.....+.....+.....+.....+...
283 C
284   150 CONTINUE
285 C
286     WRITE (*, 1120) TSTEP,NTRAJC
287     WRITE (8, 1120) TSTEP,NTRAJC
288 1120 FORMAT (//' TOTAL NUMBER OF STEPS      ',F15.0//)
289   *          ' TOTAL NUMBER OF TRAJECTORIES',I15//)
290   Write (*,1130)
291 1130 format (' End program TJI95I')
292 C
293   STOP
294 C
295 C.....+.....+.....+.....+.....+.....+...
296 C   Y(1) is R coordinate      Y(2) is THETA coordinate
297 C   Y(3) is PHI coordinate    Y(4) is V(R)
298 C   Y(5) is V(THETA)         Y(6) is V(PHI)
299 C   F(1) is R dot            F(2) is THETA dot
300 C   F(3) is PHI dot          F(4) is R dot dot
301 C   F(5) is THETA dot dot   F(6) is PHI dot dot
302 C   BR  is B(R)             BT  is B(THETA)
303 C   BP  is B(PHI)           B   is magnitude of magnetic field
304 C.....+.....+.....+.....+.....+...
305 C
306 C   ierrpt vlu  Program Format Variables printed out
307 C   IERRPT = 1  "MAIN"  1070  Input to SINGLTJ
308 C   IERRPT = 1  SINGLTJ  2000  Input to SINGLTJ
309 C   IERRPT = 2  SINGLTJ  2070  PC,BETA,KBF,RCKBETA,NSTEP,TBETA,Y,H
310 C   IERRPT = 4  SINGLTJ  2090  Y,F,ACCR,H,NSTEP
311 C   IERRPT = 3  SINGLTJ  2100  H,HCK,Y(1),DELACC,PC,NSTEP
312 C   IERRPT = 3  SINGLTJ  2110  H,HCK,Y(1),RFA,  PC,NSTEP
313 C   IERRPT = 3  SINGLTJ  2120  H,HCK,Y(1),NAMX,F(ICK),ICK,FOLD(ICK),
314 C                           ICK,PC,STEP
315 C   IERRPT = 4  SINGLTJ  2130  Y(1),DISCK,PVEL,H,HSNEK,HOLD,NSTEP
316 C   IERRPT = 4  SINGLTJ  2140  Y(1),DISCK,PVEL,H,      HOLD,NSTEP
317 C
318 C
319   END
320 SUBROUTINE GDGC (TCD, TSD)
321 C.....+.....+.....+.....+.....+...
322 C   \/ Convert from Geodetic to Geocentric coordinates
323 C   Adopted from NASA ALLMAG
324 C   GDLATD = Geodetic latitude (in degrees)
325 C   GCLATD = Geocentric latitude (in degrees)
326 C   GDCLT = Geodetic co-latitude
327 C   ERPLSQ is earth radius AT poles squared = 40408585 (km sq)
328 C   EREQSQ is earth radius AT equator squared = 40680925 (km sq)
329 C   ERADPR is earth polar radius = 6356.774733 (km)
330 C   ERADER is earth equatorial radius = 6378.160001 (km)
331 C   ERAD is earth average radius = 6371.25 (km)
332 C   ERADFL is flattening factor = 1.0/298.25
333 C   ERADFL = (ERADEQ - factor)/ERADEQ
334 C   ERECSQ is eccentricity squared = 0.00673966
335 C   ERECSQ = EREQSQ/ERPLSQ - 1.0
336 C.....+.....+.....+.....+...
337 C
338 CLast Mod 15 Jan 97 Common block SNGLR & SNGLI
339 C   Mod Feb 96 Standard reference TJ1V line check
340 C
341 C.....+.....+.....+.....+.....+...
342 C   Programmer - Don F. Smart; FORTRAN77
343 C   Note - The programming adheres to the conventional FORTRAN
344 C           default standard that variables beginning with
345 C           'i','j','k','l','m', or 'n' are integer variables
346 C           Variables beginning with "c" are character variables
347 C           All other variables are real
348 C.....+.....+.....+.....+...
349 C           Do not mix different type variables in same common block
350 C           Some computers do not allow this

```

```

351 C.....+.....+.....+.....+.....+.....+...
352 C
353     IMPLICIT INTEGER (I-N)
354     IMPLICIT REAL * 8 (A-B)
355     IMPLICIT REAL * 8 (D-H)
356     IMPLICIT REAL * 8 (O-Z)
357 C
358 C.....+.....+.....+.....+.....+.....+...
359 C
360     COMMON /WRKVLU/ F(6),Y(6),ERAD,BOMC,VEL,BR,BT,BP,B
361     COMMON /WRKTSC/ TSY2,TCY2,TSY3,TCY3
362     COMMON /TRIG/ PI,RAD,PIO2
363     COMMON /GEOID/ ERADPL, ERECSQ
364     COMMON /SNGLR/ SALT,DISOUT,GCLATD,GDLATD,GLOND,GDAZD,GDZED,
365     * RY1,RY2,RY3,RHT,TSTEP
366 C
367 C.....+.....+.....+.....+.....+.....+...
368 C
369     ERPLSQ = 40408585.0
370     EREQSQ = 40680925.0
371     ERADPL = SQRT(ERPLSQ)
372     ERECSQ = EREQSQ/ERPLSQ - 1.0
373 C
374     GDCLT = PIO2-GDLATD/RAD
375     TSGDCLT = SIN(GDCLT)
376     TCGDCLT = COS(GDCLT)
377     ONE = EREQSQ*TSGDCLT*TSGDCLT
378     TWO = ERPLSQ*TCGDCLT*TCGDCLT
379     THREE = ONE+TWO
380     RHO = SQRT(THREE)
381 C
382 C.....+.....+.....+.....+.....+.....+...
383 C     \/ Get geocentric distance from geocenter in kilometers
384 C.....+.....+.....+.....+.....+...
385 C
386     DISTKM = SQRT(SALT*(SALT+2.0*RHO)+(EREQSQ*ONE+ERPLSQ*TWO)/THREE)
387 C
388 C.....+.....+.....+.....+.....+...
389 C     TCD and TSD are sine and cosine of the angle the Geodetic vertical
390 C     must be rotated to form the Geocentric vertical
391 C.....+.....+.....+.....+...
392 C
393     TCD = (SALT+RHO)/DISTKM
394     TSD = (EREQSQ-ERPLSQ)/RHO*TCGDCLT*TSGDCLT/DISTKM
395     TCY2 = TCGDCLT*TCD-TSGDCLT*TSD
396     TSY2 = TSGDCLT*TCD+TCGDCLT*TSD
397 C
398     Y(2) = ACOS(TCY2)
399     Y(3) = GLOND/RAD
400     Y(1) = DISTKM/ERAD
401 C
402     GCLATD = (PIO2-Y(2))*RAD
403 C
404     WRITE (*,1200) GDLATD,GDCLT,TSGDCLT,TCGDCLT,ONE,TWO,THREE,RHO
405     C1200 FORMAT ('1200',8F15.5)
406     WRITE (*,1200) DISTKM,TCD,TSD,TCY2,TSY2,GCLATD
407 C
408     RETURN
409     END
410     SUBROUTINE SINGLTJ (PC,IRSLT,INDXPC,Y1GC,Y2GC,Y3GC)
411 C
412 C.....+.....+.....+.....+.....+...
413 C     Cosmic-ray trajectory calculations subroutine
414 C         calculates cosmic ray trajectory at rigidity PC
415 C.....+.....+.....+.....+.....+...
416 C     PC      = rigidity in GV
417 C     IRSLT  = trajectory result
418 C     INDXPC = index of rigidity in mv (integer)
419 C             Y1GC,Y2GC,Y3GC are initial geocentric coordinates
420 C.....+.....+.....+.....+.....+...

```

```

421 C      \/ Step size optimization & look ahead for potential BETA problems
422 C          monitor accelerating terms and reduce step length
423 C          if large increase occurs
424 C          Restart at smaller step size if BETA error occurs
425 C.....+.....+.....+.....+.....+.....+.....+...
426 C          Restrictions: Cannot run over N or S pole; will get BETA blowup
427 C.....+.....+.....+.....+.....+.....+.....+...
428 CLast Mod 17 Feb 99 if (ymax.lt.6.6) IFATE = 3
429 C Mod 18 Jan 97 Patch high latitude beta problem
430 C Mod Jan 97 High latitude step size adjust, introduce AHLT
431 C Mod Jun 96 EDIF limit set to 1.0e-5
432 C Mod Jun 96 IERRPT formats, Boundary and look ahead
433 C Mod FEB 96 standard reference TJ1V (line check 17 Feb)
434
435 C.....+.....+.....+.....+.....+.....+...
436 C          Programmer - Don F. Smart; FORTRAN77
437 C          Note - The programming adheres to the conventional FORTRAN
438 C          default standard that variables beginning with
439 C          'i','j','k','l','m', or 'n' are integer variables
440 C          Variables beginning with "c" are character variables
441 C          All other variables are real
442 C.....+.....+.....+.....+.....+.....+...
443 C          Do not mix different type variables in same common block
444 C          Some computers do not allow this
445 C.....+.....+.....+.....+.....+...
446 C
447 IMPLICIT INTEGER (I-N)
448 IMPLICIT REAL * 8 (A-B)
449 IMPLICIT REAL * 8 (D-H)
450 IMPLICIT REAL * 8 (O-Z)
451 C
452 C.....+.....+.....+.....+.....+...
453 C
454 COMMON /WRKVLU/ F(6),Y(6),ERAD,EOMC,VEL,BR,BT,BP,B
455 COMMON /WRKTSC/ TSY2,TCY2,TSY3,TCY3
456 COMMON /TRIG/ PI,RAD,PIO2
457 COMMON /GEOID/ ERADPL, ERECSQ
458 COMMON /SNGLR/ SALT,DISOUT,GCLATD,GDLATD,GLOND,GDAZD,GDZED,
459 * RY1,RY2,RY3,RHT,TSTEP
460 COMMON /SNGLI/ LIMIT,NTRAJC,IERRPT
461 C
462 C.....+.....+.....+.....+.....+...
463 C
464 DIMENSION P(6),Q(6),R(6),S(6),YB(6),FOLD(6),YOLD(6)
465 C
466 C.....+.....+.....+.....+.....+...
467 C
468 CHARACTER*1 CF,CR
469 CHARACTER*6 CNAME
470 C
471 DATA CF,CR / 'F','R'/
472 DATA CNAME / ' I95 '/                                ###
473 C
474 C.....+.....+.....+.....+...
475 C
476 IF (IERRPT.GT.0) WRITE (16,2000) PC,INDXPC,RY1,RY2,RY3
477 2000 FORMAT (' SINGLTJ ',F8.3,I8,3F8.4)
478 C
479 BETAST = 2.0
480 LSTEP = 0
481 KBF = 0
482 C
483 C.....+.....+.....+.....+.....+...
484 C          \/ Runge-Kutta constants
485 C.....+.....+.....+.....+.....+...
486 C
487 RC1O6 = 1.0/6.0
488 SR2 = SQRT(2.0)
489 TMS2O2 = (2.0-SR2)/2.0
490 TPS2O2 = (2.0+SR2)/2.0

```

```
491 C
492 C.....+.....+.....+.....+.....+.....+...
493 C     \/ Initialize Runge-Kutta variables to zero
494 C.....+.....+.....+.....+.....+...
495 C
496     100 DO 110 I = 1, 6
497         YB(I) = 0.0
498         S(I) = 0.0
499         R(I) = 0.0
500         Q(I) = 0.0
501         P(I) = 0.0
502         F(I) = 0.0
503     110 CONTINUE
504 C
505     NMAX = 0
506     NMIN = 0
507     NSTEP = 0
508     NSTEPT = 0
509 C
510     TAU = 0.0
511     TU100 = 0.0
512     YMAX = RY1
513 C
514 C.....+.....+.....+.....+.....+.....+...
515 C     \/ Define initial point at start of trajectory
516 C.....+.....+.....+.....+.....+...
517 C
518     Y(1) = RY1
519     Y(2) = RY2
520     Y(3) = RY3
521     GRNDKM = (ERADPL/SQRT(1.0-ERECSQ*TSY2SQ))
522     Y10 = (RHT+GRNDKM)/ERAD
523     R120KM = (ERAD+120.0)/ERAD
524 C
525 C.....+.....+.....+.....+.....+...
526 C     Rigidity = momentum/charge
527 C             use oxygen 16 as reference isotope
528 C             Constants used from Handbook of Physics (7-170)
529 C                     1 amu = 0.931141 GeV
530 C.....+.....+.....+.....+...
531 C
532     ANUC = 16.0
533     ZCHARGE = 8.0
534 C
535     EMCSQ = 0.931141
536     TENG = SQRT((PC*ZCHARGE)**2+(ANUC*EMCSQ)**2)
537     EOMC = -8987.566297*ZCHARGE/TENG
538     GMA = SQRT(((PC*ZCHARGE)/(EMCSQ*ANUC))**2+1.0)
539     BETA = SQRT(1.0-1.0/(GMA*GMA))
540     PVEL = VEL*BETA
541     HMAX = 1.0/PVEL
542     DISCK = DISOUT - 1.1*HMAX*PVEL
543 C
544 C.....+.....+.....+.....+.....+...
545 C     \/ Set max step length ("HMAX") to 1 earth radii
546 C             PVEL is particle velocity in earth radii per second
547 C             DISCK is check for approaching termination boundary
548 C                     (within 1.1 steps)
549 C.....+.....+.....+.....+...
550 C
551     EDIF = BETA*1.0E-4
552     if (edif.lt.1.0-5) edif = 1.0e-5
553     if (beta.lt.0.1)   edif = 1.0e-4
554 C
555     Y(4) = BETA*Y1GC
556     Y(5) = BETA*Y2GC
557     Y(6) = BETA*Y3GC
558 C
559     azd = gdazz
560     zed = gdzed
```

```

561      IAZ = AZD+0.01
562      IZE = ZED+0.01
563      C
564      C.....+.....+.....+.....+.....+.....+...
565      C  \/ Set HSTART to about 1 % of the time to complete one gyro-radius
566      C          in a 1 Gauss field
567      C          H = [(2.0*PI*33.333*PC)/(BETA*C)]/0.01
568      C          if restart after BETA error, set HCK to small value
569      C          Introduce AHLT to control step size at high lat (beta problem)
570      C          HCK - reduce step size when large acceleration
571      C          HOLD - last step size used
572      C          HCNG - only allow 20% max growth in step size
573      C          HSNEK - attempt to approach boundary quickly
574      C          Problem at z=90 at high lat
575      C          add zen angle in deg to reduce first step
576      C.....+.....+.....+.....+.....+...
577      C
578      PTCY2 = ABS(TCY2)
579      AHLT = (1.0 + PTCY2)**2
580      HSTART = 6.0E-6*PC/(BETA*AHLT + ZED*PTCY2)
581      IF (HSTART.LT.1.0E-6) HSTART = 1.0E-6
582      HOLD = HSTART
583      HCK = HSTART
584      HCNG = HSTART
585      C
586      C      WRITE (16, 2010) HMAX,HOLD,HCK,HCNG,Y(4),Y(5),Y(6),PVEL, NSTEP
587      C2010 FORMAT (' 2010 ',18X, 4F9.6, 3F9.4, F9.4,9X,15X,I6)
588      C
589      C.....+.....+.....+.....+.....+...
590      C      Start Runge-Kutta
591      C      \\\//\//\//\//\//\//
592      C      \/\//\//\//\//\/
593      C      \/\//\//\//
594      C      \/\\
595      C      \
596      C.....+.....+.....+.....+.....+...
597      C      Change in step size criteria, Aug 97
598      C          remove cos VxB step size, causes problems in tight loops
599      C          step size is now only a function of B and BETA
600      C.....+.....+.....+.....+...
601      C
602      130 IF (HCK.LT.1.0E-6) HCK = 1.0E-6
603      CALL FGRAD
604      HB = 1.6E-5*PC/(B*BETA)
605      H = HB/BETAST
606      C
607      IF (KBF.GT.0) H=H/(FLOAT(KBF*2))
608      IF (H.GT.HMAX) H = HMAX
609      IF (H.GT.HCNG) H = HCNG
610      IF (H.GT.HCK) H = HCK
611      C
612      DO 140 I = 1, 6
613          S(I) = H*F(I)
614          P(I) = 0.5*S(I)-Q(I)
615          YB(I) = Y(I)
616          Y(I) = Y(I)+P(I)
617          R(I) = Y(I)-YB(I)
618          Q(I) = Q(I)+3.0*R(I)-0.5*S(I)
619      140 CONTINUE
620      C
621      CALL FGRAD
622      C
623      DO 150 I = 1, 6
624          S(I) = H*F(I)
625          P(I) = TMS2O2*(S(I)-Q(I))
626          YB(I) = Y(I)
627          Y(I) = Y(I)+P(I)
628          R(I) = Y(I)-YB(I)
629          Q(I) = Q(I)+3.0*R(I)-TMS2O2*S(I)
630      150 CONTINUE

```

```

631   C
632   CALL FGRAD
633   C
634   DO 160 I = 1, 6
635     S(I) = H*F(I)
636     P(I) = TPS2O2*(S(I)-Q(I))
637     YB(I) = Y(I)
638     Y(I) = Y(I)+P(I)
639     R(I) = Y(I)-YB(I)
640     Q(I) = Q(I)+3.0*R(I)-TPS2O2*S(I)
641 160 CONTINUE
642   C
643   CALL FGRAD
644   C
645   DO 170 I = 1, 6
646     S(I) = H*F(I)
647     P(I) = RC1O6*(S(I)-2.0*Q(I))
648     YB(I) = Y(I)
649     Y(I) = Y(I)+P(I)
650     R(I) = Y(I)-YB(I)
651     Q(I) = Q(I)+3.0*R(I)-0.5*S(I)
652 170 CONTINUE
653   C.....+.....+.....+.....+.....+.....+...
654   C.....+.....+.....+.....+.....+.....+...
655   C.....+.....+.....+.....+.....+.....+...
656   C.....+.....+.....+.....+.....+.....+...
657   C.....+.....+.....+.....+.....+.....+...
658   C.....+.....+.....+.....+.....+.....+...
659   C.....+.....+.....+.....+.....+.....+...
660   C.....+.....+.....+.....+.....+.....+...
661   C.....+.....+.....+.....+.....+.....+...
662   C.....+.....+.....+.....+.....+.....+...
663   C.....+.....+.....+.....+.....+.....+...
664   NSTEP = NSTEP+1
665   NSTEPT = NSTEPT + 1
666   TAU = TAU+H
667   HOLD = H
668   HCNG = H*1.2
669   HCK = HCNG
670   C.....+.....+.....+.....+.....+.....+...
671   C.....+.....+.....+.....+.....+.....+...
672   C.....+.....+.....+.....+.....+.....+...
673   C.....+.....+.....+.....+.....+.....+...
674   C.....+.....+.....+.....+.....+.....+...
675   C.....+.....+.....+.....+.....+.....+...
676   C.....+.....+.....+.....+.....+.....+...
677   C.....+.....+.....+.....+.....+.....+...
678   C.....+.....+.....+.....+.....+.....+...
679   C.....+.....+.....+.....+.....+.....+...
680   C.....+.....+.....+.....+.....+.....+...
681   C.....+.....+.....+.....+.....+.....+...
682   C.....+.....+.....+.....+.....+.....+...
683   C.....+.....+.....+.....+.....+.....+...
684   C.....+.....+.....+.....+.....+.....+...
685   C.....+.....+.....+.....+.....+.....+...
686   C.....+.....+.....+.....+.....+.....+...
687   C.....+.....+.....+.....+.....+.....+...
688   IF (Y(1).LT.R120KM) THEN
689     TSY2SQ = SIN(Y(2))**2
690     GRNDKM = (ERADPL/SQRT(1.0-ERECSQ*TSY2SQ))
691     R100KM = (100.0+GRNDKM)/ERAD
692     R120KM = (120.0+GRNDKM)/ERAD
693     IF (Y(1).LT.R100KM) TU100 = TU100+H
694     PSALT = Y(1)*ERAD-GRNDKM
695     Y10 = (RHT+GRNDKM)/ERAD
696   C.....+.....+.....+.....+.....+.....+...
697   IF (NSTEP.GT.5) THEN
698     IF (Y(1).LT.Y10.OR.PSALT.LE.0.0) THEN
699       IF (IERRPT.GT.2) WRITE (16, 2045) PSALT, Y(1), Y10
700       IRT = -1

```

```

701          GO TO 260
702      ENDIF
703      ENDIF
704      ENDIF
705 2045 FORMAT (' 2045 PSALT,Y(1),Y10',F10.6,1PE14.6,E14.6)
706 C
707 C.....+.....+.....+.....+.....+.....+...
708 C  \/ Begin error checks
709 C    (1) Check for unacceptable changes in BETA
710 C.....+.....+.....+.....+.....+.....+...
711 C
712 RCKBETA = SQRT(Y(4)*Y(4)+Y(5)*Y(5)+Y(6)*Y(6))
713 TBETA = BETA-RCKBETA
714 IF (ABS(TBETA).GT.EDIF) THEN
715   KBF = KBF+1
716   BETAST = BETAST + AHLT
717   EDIF = 2.0*EDIF
718   IF (RCKBETA.GT.(1.0+EDIF)) THEN
719     BETAST = BETAST+FLOAT(KBF)*(1.0+AHLT)
720     WRITE (*,2050) KBF,BETA,RCKBETA
721     WRITE (*,2060) Y,H,PC,NSTEP
722     WRITE (16,2050) KBF,BETA,RCKBETA
723     WRITE (16,2060) Y,H,PC,NSTEP
724   ENDIF
725 C
726   WRITE (16,2070) PC,BETA,KBF,RCKBETA,NSTEP,TBETA,Y,H
727   WRITE (*,2070) PC,BETA,KBF,RCKBETA,NSTEP,TBETA,Y,H
728 C
729 C.....+.....+.....+.....+.....+...
730 C  \/ Check for irrecoverable beta error
731 C    if KBF > 4, set fate to failed and start next rigidity
732 C.....+.....+.....+.....+...
733 C
734 IF (KBF.lt.5) THEN
735   GO TO 100
736 ELSE
737   IRT = 0
738   PATH = -PVEL*TAU
739   ISALT = SALT+0.0001
740   WRITE (*,2080)
741   GO TO 280
742 ENDIF
743 ENDIF
744 C
745 2050 FORMAT (' 2050 ','KBF=',I5,5X,' error, the velocity of the ',
746 *           'particle (BETA) has exceeded the velocity of light',
747 *           'BETA at start of trajectory was ',F10.7/
748 *           'BETA now equals ',F10.7/
749 *           'reduce step size and try again   ')
750 2060 FORMAT (' 2060 ','Y,H,PC,NSTEP=',8F12.6,I10)
751 2070 FORMAT (' 2070 ','ERROR, Particle BETA changed;',
752 *           ' PC = ',F10.6,' GV',
753 *           ' 6x, Beta at start was ',F10.7,17X,'KBF=',I6/
754 *           ' 6x, Beta new equals ',F10.7,10X,'step number',I6/
755 *           ' 6x, Beta difference ',F10.7/
756 *           ' 6x, step length reduced and trajectory recalculated ',
757 *           ' 6x,'Y=',6F12.6,6X,' H=',F15.7)
758 2080 FORMAT (' 2080 ','Irrecoverable BETA error problem'/6X,
759 *           ' Set fate to Failed & make path length negative'/6X,
760 *           ' Terminate this trajectory & continue with program')
761 C
762 C.....+.....+.....+.....+.....+...
763 C  \/ Continue status checks
764 C    (2) Compute composite acceleration
765 C.....+.....+.....+.....+.....+...
766 C
767 ACCER = SQRT(F(4)*F(4)+F(5)*F(5)+F(6)*F(6))
768 C
769 IF (IERRPT.GT.3) WRITE (16,2090) Y, F, ACCER, H, NSTEP
770 2090 FORMAT (' Y,F,A,H ',F7.4,5F7.3,1X,1PE8.1,5E8.1,1X,E8.1, E9.2,I6)

```

```

771 C      ....+.....+.....+.....+.....+.....+...
772 C      \/ Continue status checks, make adjustment latitude dependant
773 C          (3) Monitor change in composite acceleration
774 C              If composite acceleration (new-old) change > 5
775 C              If composite acceleration (new/old) ratio > 2
776 C                  change step size to a smaller value
777 C      ....+.....+.....+.....+.....+...
778 C
779 C
780 IF (NSTEP.GE.2) THEN
781     IF (ACCR.GT.ACCLD) THEN
782         DELACC = ACCR-ACCLD
783         IF (DELACC.GT.5.0) THEN
784             HCK = HCK/(1.0+AHLT)
785             IF (IERRPT.GT.2) WRITE (16,2100)
786             *           H,HCK,Y(1),DELACC,PC,NSTEP
787             RFA = ACCR/ACCLD
788             IF (RFA.GT.2.0) THEN
789                 HCK = HCK/(1.0+AHLT)
790                 IF (IERRPT.GT.2) WRITE (16,2110)
791                 *           H,HCK,Y(1),RFA,PC,NSTEP
792             ENDIF
793         ENDIF
794     ENDIF
795 C
796 2100 FORMAT (' 2100 ','H-REDUCE',2X,'H=',F8.6,2X,'HCK=',F8.6,2X,
797     *           'Y(1)=',F7.4,2X,'DELACC=',F6.2,4X,'PC=',F8.3,4X,'NSTEP=',I8)
798 2110 FORMAT (' 2110 ','H-REDUCE',2X,'H=',F8.6,2X,'HCK=',F8.6,2X,
799     *           'Y(1)=',F7.4,4X,'RFA=',F6.2,4X,'PC=',F8.3,4X,'NSTEP=',I8)
800 C      ....+.....+.....+.....+.....+.....+...
801 C      \/ Continue status checks, make adjustment latitude dependant
802 C          (4) Monitor change in acceleration components
803 C              If change in any acceleration component is more than
804 C                  a factor of 3, reduce step length
805 C      ....+.....+.....+.....+...
806 C
807 C
808 DO 200 ICK = 4, 6
809     AFOLD = ABS(FOLD(ICK))
810     IF (AFOLD.GT.3.0) THEN
811         RFCK = ABS(F(ICK)/AFOLD)
812         IF (RFCK.GT.3.0) THEN
813             HCK = HCK/(1.0+AHLT)
814             IF (IERRPT.GT.2) THEN
815                 WRITE (16,2120) H,HCK,Y(1),NMAX,ICK,F(ICK),
816                 ICK,FOLD(ICK),PC,NSTEP
817             &
818             ENDIF
819         ENDIF
820     200 CONTINUE
821     ENDIF
822 C
823 2120 FORMAT (' 2120 ','H-reduce',2X,'H=',F8.6,2X,'HCK=',F8.6,2X,
824     *           'Y(1)=',F7.4,2X,'NAMX=',I4,2X,'F(',I1,')=',F6.2,2X,
825     *           'FOLD(',I1,')=',F6.2,2X,'PC=',F6.3,2X,'NSTEP=',I6)
826 C
827     ACCOLD = ACCR
828 C
829 C      ....+.....+.....+.....+.....+...
830 C      \/ Error checks complete
831 C
832 C      \/ Find if a max or a min has occurred
833 C      ....+.....+.....+.....+...
834 C
835 IF (NSTEP.GT.1) THEN
836     IF (YOLD(4).LE.0.0.AND.Y(4).GT.0.0) NMIN = NMIN+1
837     IF (YOLD(4).GE.0.0.AND.Y(4).LT.0.0) NMAX = NMAX+1
838 ENDIF
839 C
840 IF (Y(1).GT.YMAX) YMAX = Y(1)

```

```
841 C
842 C.....+.....+.....+.....+.....+.....+...
843 C   \/ Check for termination conditions
844 C     Allowed - radial distance exceeded disout
845 C     Failed - number of steps exceeded
846 C     Re-entrant - trajectory is below "top" of atmosphere
847 C
848 C.....+.....+.....+.....+.....+...
849 C   \/ (1) Check for step limit exceeded
850 C.....+.....+.....+.....+...
851 C
852 IF (NSTEP.GE.LIMIT) THEN
853   IRT = 0
854   GO TO 260
855 ENDIF
856 C
857 C.....+.....+.....+.....+...
858 C   \/ (2) Check if y(1) within 1.1 max step lengths of disout.
859 C     if so, reduce step size and
860 C       approach boundary at smaller step
861 C.....+.....+.....+...
862 C
863 IF (Y(1).GT.DISCK) THEN
864   DISTR = ABS(DISOUT - Y(1))
865   HSNEK = DISTR/PVEL
866   HCNG = HCNG/2.0
867   HCK = HCK/2.0
868   IF (HSNEK .LT. HCNG) HCNG = HSNEK
869   IF (HSNEK .LT. HCK) HCK = HSNEK
870   DISCK = DISOUT - DISTR/2.0
871   IF (DISCK.GE.DISOUT) THEN
872     DISCK = 24.999
873     GO TO 210
874   ENDIF
875   IF (H.LT.1.0E-5 .OR. HCK.LT.1.0E-5 .OR. HCNG.LT.1.0E-5) THEN
876     H = 1.0E-5
877     HCK = 1.0E-5
878     HCNG = 1.0E-5
879   ENDIF
880 C
881 IF (IERRPT.GT.3) WRITE (16,2130) Y(1),DISCK,PVEL,H,HSNEK,NSTEP
882 C
883 210 IF (Y(1).GT.DISOUT) THEN
884   IF (H.LE.1.0E-5) THEN
885     IRT = 1
886     GO TO 260
887   ENDIF
888   TAU = TAU - H
889   DO 220 I = 1, 6
890     Y(I) = YOLD(I)
891     F(I) = FOLD(I)
892   220 CONTINUE
893   endif
894   GO TO 130
895 ENDIF
896 2130 FORMAT (' 2130 ',2X,'Y(1),DISCK,PVEL,H,HSNEK',
897 *           4X,1PE12.6,4X,E12.6,4X,E12.6,4X,2E9.2,22X,I6)
898 C
899 C.....+.....+.....+.....+...
900 C   \/ Have penetrated boundary if you are here.
901 C     if large step size, go back one step and
902 C       reduce step length (and adjust "TAU")
903 C.....+.....+.....+...
904 C
905 230 IF (Y(1).GT.DISOUT) THEN
906 C
907   IF (IERRPT.GT.3) WRITE (16, 2140) y(1),disck,pvel,H,nstep
908 C
909   if (h.lt.1.0e-5 .or. hck.lt.1.0e-5 .or. hcng.lt.1.0e-5) then
910     IRT = 1
```

```

911      go to 260
912      else
913          hck = hck/2.0
914          hcng = hcng/2.0
915          TAU = TAU - H
916          DO 240 I = 1, 6
917              Y(I) = YOLD(I)
918              F(I) = FOLD(I)
919      240      CONTINUE
920      GO TO 130
921      ENDIF
922      ENDIF
923 C
924     2140 FORMAT (' 2140 ',2x,'y(1),disck,pvel,H',
925             *           4x,1pe12.6,4x,e12.6,4x,e12.6,4x,e9.2,27x,i6)
926 C
927 C.....+.....+.....+.....+.....+.....+.....+...
928 C   \/ Store values of Y and F as FOLD & YOLD
929 C.....+.....+.....+.....+.....+.....+...
930 C
931     DO 250 I = 1, 6
932         YOLD(I) = Y(I)
933         FOLD(I) = F(I)
934     250 CONTINUE
935 C
936     GO TO 130
937 C
938 C.....+.....+.....+.....+.....+.....+...
939 C*****+*****+*****+*****+*****+*****+*****
940 C           *****           *****           *****
941 C           *****           *****           *****
942 C           *****
943 C           TRAJECTORY COMPLETE IF YOU ARE HERE
944 C.....+.....+.....+.....+.....+...
945 C
946     260 CONTINUE
947 C
948     IF (Y(1).GE.DISOUT)  IRT = 1
949     PATH = PVEL*TAU
950     ISALT = SALT+0.0001
951     LSTEP = BETAST - 1.9
952 C
953 C.....+.....+.....+.....+.....+...
954 C   \/ Write out results
955 C     IRT    +1    ALLOWED      (FATE = 0)
956 C     IRT    0     FAILED       (FATE = 2)
957 C     IRT   -1    RE-ENTRANT  (FATE = 1)
958 C.....+.....+.....+.....+...
959 C
960     IF (IRT.GT.0) THEN
961         TCY2 = COS(Y(2))
962         TSY2 = SIN(Y(2))
963         YDAS5 = Y(5)*TCY2+Y(4)*TSY2
964         ATRG1 = Y(4)*TCY2-Y(5)*TSY2
965         ATRG2 = SQRT(Y(6)*Y(6)+YDAS5*YDAS5)
966         FASLAT = 0.0
967         IF (ATRG1.NE.0.0.AND.ATRG2.NE.0.0) FASLAT =
968             ATAN2(ATRG1,ATRG2)*RAD
969         *
970         FASLON = Y(3)*RAD
971         IF (Y(6).NE.0.0.AND.YDAS5.NE.0.0) FASLON = (Y(3)+ATAN2(Y(6),
972                                         YDAS5))*RAD
973         *
974         IF (FASLON.LT.0.0)    FASLON = FASLON+360.0
975         IF (FASLON.GT.360.0) FASLON = FASLON-360.0
976 C
977         WRITE (8,2150) GDLATD,GCLATD,GLOND,IZE,IAZ,PC,FASLAT,FASLON,
978             PATH,NMAX,NSTEP,TU100,YMAX,LSTEP,SALT,CNAME
979 C
980         IFATE = 0
981         WRITE (7,2160) GDLATD,GLOND,PC,ZED,AZD,ISALT,FASLAT,FASLON,
982             NSTEP,IFATE,CNAME
983

```

```

981      ENDF
982 2150 FORMAT (2F7.2,F9.2,I5,I4,F10.3,2F8.2,F11.5,I4,I7,F9.5,F9.4,
983      *          I4,F11.1,1X,A6,13X)
984 2160 FORMAT (F7.2,F8.2,F9.3,2F6.1,I7,F7.2,F8.2,I7,3X,I3,3X,A6)
985 C
986      IF (IRT.LT.0) THEN
987          RENLAT = (PIO2-Y(2))*RAD
988          RENLON = Y(3)*RAD
989 C
990      WRITE (8,2170) GDLATD,GCLATD,GLOND,IZE,IAZ,PC,CR,CR,PATH,NMAX
991      *          ,NSTEP,TU100,YMAX,LSTEP,SALT,CNAME,RENLAT,RENLON
992 C
993      IFATE = 1
994      WRITE (7,2180) GDLATD,GLOND,PC,ZED,AZD,ISALT,NSTEP,IFATE,CNAME
995      ENDIF
996 2170 FORMAT (2F7.2,F9.2,I5,I4,F10.3,5X,A1,2X,5X,A1,2X,F11.5,I4,I7,
997      *          F9.5,F9.4,I4,F11.1,1X,A6,F6.1,F7.1)
998 2180 FORMAT (F7.2,F8.2,F9.3,2F6.1,I7,4X,'R',7X,'R',I9,3X,I3,3X,A6)
999 C
1000 280 IF (IRT.EQ.0) THEN
1001 C
1002      WRITE (8,2190) GDLATD,GCLATD,GLOND,IZE,IAZ,PC,CF,CF,PATH,
1003      *          NMAX,NSTEP,TU100,YMAX,LSTEP,SALT,CNAME
1004 C
1005      IFATE = 2
1006      IF (YMAX.LT.6.6) IFATE = 3
1007      WRITE (7,2200) GDLATD,GLOND,PC,ZED,AZD,ISALT,NSTEP,IFATE,CNAME
1008      ENDIF
1009 2190 FORMAT (2F7.2,F9.2,I5,I4,F10.3,5X,A1,2X,5X,A1,2X,F11.5,I4,I7,
1010      *          F9.5,F9.5,I4,F11.1,1X,A6,13X)
1011 2200 FORMAT (F7.2,F8.2,F9.3,2F6.1,I7,4X,'F',7X,'F',I9,3X,I3,3X,A6)
1012 C
1013      NTRAJC = NTRAJC+1
1014      TSTEP = TSTEP+FLOAT(NSTEP)
1015 C
1016 C ...+.....+.....+.....+.....+.....+.....+...
1017 C \/ Comment out to reduce IO
1018 C ...+.....+.....+.....+.....+.....+...
1019 C
1020 C     WRITE (*,2210) PC, ZED, AZD, NSTEP, IFATE
1021 C2210 FORMAT (1H+, 22X, 3F7.2,7x,2I6)
1022 C
1023      IRSLT = IRT
1024      RETURN
1025      END
1026      SUBROUTINE FGRAD
1027 C
1028 C.....+.....+.....+.....+.....+.....+...
1029 C     Mod Feb 96 standard reference TJ1V (line check 17 Feb)
1030 C     Mod 27 Jan 1999 Change MAGNEW to NEWMAG95           ###
1031 C.....+.....+.....+.....+.....+.....+...
1032 C     Programmer - Don F. Smart; FORTRAN77
1033 C     Note - The programming adheres to the conventional FORTRAN
1034 C             default standard that variables beginning with
1035 C             'i','j','k','l','m', or 'n' are integer variables
1036 C             Variables beginning with "c" are character variables
1037 C             All other variables are real
1038 C.....+.....+.....+.....+.....+...
1039 C             Do not mix different type variables in same common block
1040 C             Some computers do not allow this
1041 C.....+.....+.....+.....+...
1042 C
1043      IMPLICIT INTEGER (I-N)
1044      IMPLICIT REAL * 8(A-B)
1045      IMPLICIT REAL * 8(D-H)
1046      IMPLICIT REAL * 8(O-Z)
1047 C
1048 C.....+.....+.....+.....+...
1049 C
1050      COMMON /WRKVLU/ F(6),Y(6),ERAD,EOMC,VEL,BR,BT,BP,B

```

```

1051      COMMON /WRKTSC/ TSY2,TCY2,TSY3,TCY3
1052      C
1053      C.....+.....+.....+.....+.....+.....+...
1054      C
1055      F(1) = VEL*Y(4)
1056      F(2) = VEL*Y(5)/Y(1)
1057      TSY2 = SIN(Y(2))
1058      TCY2 = COS(Y(2))
1059      F(3) = VEL*Y(6)/(Y(1)*TSY2)
1060      SQY6 = Y(6)*Y(6)/Y(1)
1061      Y5OY1 = Y(5)/Y(1)
1062      TAY2 = TSY2/TCY2
1063      CALL MAGNEW95
1064      F(4) = EOMC*(Y(5)*BP-Y(6)*BT)+VEL*(Y(5)*Y5OY1+SQY6)
1065      F(5) = EOMC*(Y(6)*BR-Y(4)*BP)+VEL*(SQY6/TAY2-Y5OY1*Y(4))
1066      F(6) = EOMC*(Y(4)*BT-Y(5)*BR)-VEL*((Y5OY1*Y(6))/TAY2+Y(4)*Y(6) /
1067      * Y(1))
1068      RETURN
1069      C
1070      C.....+.....+.....+.....+.....+.....+...
1071      C      Y(1) is R coordinate          Y(2) is THETA coordinate
1072      C      Y(3) is PHI coordinate        Y(4) is V(R)
1073      C      Y(5) is V(THETA)           Y(6) is V(PHI)
1074      C      F(1) is R dot             F(2) is THETA dot
1075      C      F(3) is PHI dot          F(4) is R dot dot
1076      C      F(5) is THETA dot dot    F(6) is PHI dot dot
1077      C      BR is B(R)              BT is B(THETA)
1078      C      BP is B(PHI)            B is magnitude of magnetic field
1079      C.....+.....+.....+.....+.....+...
1080      C
1081      END
1082      SUBROUTINE MAGNEW95
1083      C
1084      C.....+.....+.....+.....+.....+...
1085      C      Compute Magnetic field
1086      C      Derived from NASA (NSSDC) routine NEWMAG version of December 1965
1087      C      modified for 10 order field
1088      C      Coefficients for IGRF 1995 loaded into this subroutine
1089      C      Coefficients obtained from program CNGMAGN
1090      C.....+.....+.....+.....+...
1091      C      CLast Mod 27 Jan 1999 IGRF 95 coefficients
1092      C      Mod Feb 1996 standard reference TJ1V (line check 17 Feb)
1093      C      Mod Nov 1980 for arguments in labeled common
1094      C.....+.....+.....+...
1095      C.....+.....+.....+...
1096      C      Programmer - Don F. Smart; FORTRAN77
1097      C      Note - The programming adheres to the conventional FORTRAN
1098      C      default standard that variables beginning with
1099      C      'i','j','k','l','m', or 'n' are integer variables
1100      C      Variables beginning with "c" are character variables
1101      C      All other variables are real
1102      C.....+.....+.....+...
1103      C      Do not mix different type variables in same common block
1104      C      Some computers do not allow this
1105      C.....+.....+.....+...
1106      C
1107      IMPLICIT INTEGER (I-N)
1108      IMPLICIT REAL * 8(A-B)
1109      IMPLICIT REAL * 8(D-H)
1110      IMPLICIT REAL * 8(O-Z)
1111      C
1112      C.....+.....+.....+...
1113      C
1114      COMMON /WRKVLU/ F(6),Y(6),ERAD,EOMC,VEL,BR,BT,BP,B
1115      COMMON /WRKTSC/ TSY2,TCY2,TSY3,TCY3
1116      C
1117      C.....+.....+...
1118      C
1119      DIMENSION G(11,11),BM(11)
1120      C

```

```

1121 C.....+.....+.....+.....+.....+.....+...
1122 C
1123 C      \/ Load in data constants if this is the first time called
1124 C          otherwise, skip to evaluation of magnetic field
1125 C          designed to drop high order terms if contribution
1126 C              would be less than "BERR"
1127 C          also designed so the maximum order of expansion
1128 C              can be specified
1129 C.....+.....+.....+.....+.....+...
1130 C
1131     IF (JDATA.EQ.77) GO TO 120
1132 C
1133 C.....+.....+.....+.....+.....+...
1134 C      Gauss normalized Schmidt coefficients ordered for fast computation
1135 C
1136 Cards for FORTRAN
1137 C      1995.00      Coef in CNGMAG format           IGRF95
1138     DATA(G(N, 1),N=1,11)/ 0.000000E+00
1139     c, 0.296820E+05, 0.329550E+04, -0.332250E+04, -0.411687E+04
1140     c, 0.165375E+04, -0.952875E+03, -0.209137E+04, -0.120656E+04
1141     c, -0.379844E+03,
1142     c 0.541277E+03/
1143     DATA(G(N, 2),N=1,11)/ -0.531800E+04
1144     c, 0.178900E+04, -0.532432E+04, 0.694430E+04, -0.432758E+04
1145     c, -0.357864E+04, -0.120980E+04, 0.237646E+04, -0.268125E+03
1146     c, -0.114663E+04,
1147     c 0.973144E+03/
1148     DATA(G(N, 3),N=1,11)/ 0.408071E+04
1149     c, 0.368061E+03, -0.145925E+04, -0.241868E+04, -0.113872E+04
1150     c, -0.182140E+04, -0.971375E+03, -0.289608E+02, 0.560824E+02
1151     c, -0.108650E+03,
1152     c -0.421384E+03/
1153     DATA(G(N, 4),N=1,11)/ 0.805270E+03
1154     c, -0.584820E+03, 0.320971E+03, -0.607948E+03, 0.880585E+03
1155     c, 0.574158E+03, 0.171361E+04, -0.593873E+03, 0.372776E+03
1156     c, 0.995794E+03,
1157     c 0.826402E+03/
1158     DATA(G(N, 5),N=1,11)/ -0.144990E+04
1159     c, 0.907844E+03, -0.204982E+03, 0.222593E+03, -0.857832E+02
1160     c, 0.370495E+03, -0.109137E+02, -0.493957E+02, 0.374307E+03
1161     c, -0.507382E+03,
1162     c 0.233742E+03/
1163     DATA(G(N, 6),N=1,11)/ -0.447330E+03
1164     c, -0.120658E+04, 0.715344E+03, 0.141986E+03, -0.694545E+02
1165     c, 0.182406E+02, -0.395558E+02, -0.493957E+02, -0.593223E+02
1166     c, 0.134764E+03,
1167     c -0.295662E+03/
1168     DATA(G(N, 7),N=1,11)/ 0.302450E+03
1169     c, -0.115071E+04, -0.667509E+03, 0.311041E+03, -0.930726E+01
1170     c, -0.188074E+02, 0.631392E+02, -0.242182E+02, -0.343261E+02
1171     c, 0.347959E+02,
1172     c -0.123960E+03/
1173     DATA(G(N, 8),N=1,11)/ 0.273116E+04
1174     c, 0.724020E+03, -0.614352E+02, -0.271676E+03, -0.987915E+02
1175     c, 0.557020E+02, 0.194178E+01, 0.129452E+01, 0.000000E+00
1176     c, -0.527347E+02,
1177     c -0.200432E+02/
1178     DATA(G(N, 9),N=1,11)/ -0.804375E+03
1179     c, 0.112165E+04, -0.289937E+03, 0.561461E+03, -0.177967E+03
1180     c, -0.686523E+02, 0.426161E+02, 0.626707E+01, 0.438695E+01
1181     c, 0.000000E+00,
1182     c -0.245478E+02/
1183     DATA(G(N, 10),N=1,11)/ 0.242067E+04
1184     c, -0.162975E+04, -0.912811E+03, 0.394630E+03, 0.235837E+03
1185     c, -0.156581E+03, -0.527347E+02, 0.206718E+02, -0.609049E+00
1186     c, 0.365430E+01,
1187     c -0.796435E+01/
1188     DATA(G(N, 11),N=1,11)/ -0.486572E+03
1189     c, -0.210692E+03, -0.495841E+03, -0.701225E+03, 0.295662E+03
1190     c, 0.000000E+00, 0.400864E+02, -0.245478E+02, 0.265478E+01

```

```

1191      c, 0.356177E+01,
1192      c 0.000000E+00/
1193      DATA JMAG/ 0/,MGNMAX/ 11/,GSUM/ -0.885846E+05/
1194      DATA BM/ 0.100078E+06
1195      c, 0.100078E+06, 0.396633E+05, 0.253449E+05, 0.134069E+05
1196      c, 0.684114E+04, 0.360156E+04, 0.197007E+04, 0.913524E+03
1197      c, 0.489229E+03,
1198      c 0.152640E+03/
1199      C 1995.00 Coef in CNGMAG format                                IGRF95
1200      C ****
1201      C * The array G contains Gauss normalized Schmidt coefficients
1202      C * the array G contains both the G and H coefficients
1203      C * G(1,1) = 0.0
1204      C *Schmidt G(N,M) corresponds to -G(NN+1,MM+1) Gauss normalized coef
1205      C *Schmidt H(N,M) corresponds to -G( MI ,NN+1) Gauss normalized coef
1206      C *
1207      C * where MI = M
1208      C ****
1209      C
1210      IF (GMSUM.EQ.0) GO TO 110
1211      P22 = 0.
1212      BERR = 0.0001
1213      AR = 0.
1214      DO 100 L = 1, MGNMAX
1215          DO 100 M = 1, MGNMAX
1216              AR = AR+1.
1217              P22 = P22+AR*G(M,L)
1218      100 CONTINUE
1219      GMSUM = (GMSUM-P22)/GMSUM
1220      C.....+.....+.....+.....+.....+.....+...
1221      C**** \/ Note following print and stop statements
1222      C.....+.....+.....+.....+.....+.....+...
1223      C
1224      C
1225      IF (ABS(GMSUM).GT.1.E-4) THEN
1226          WRITE (*, 2200) GMSUM
1227          WRITE (7, 2200) GMSUM
1228          WRITE (8, 2200) GMSUM
1229          STOP
1230      ENDIF
1231      2200 FORMAT (' DATA WRONG IN MAGNEW',E15.6)
1232      C
1233      110 CONTINUE
1234      C
1235      GMSUM = 0.
1236      JDATA = 77
1237      C
1238      120 CONTINUE
1239      P21 = TCY2
1240      P22 = TSY2
1241      AR = 1.0/Y(1)
1242      C
1243      C.....+.....+.....+.....+.....+...
1244      C \/ N= 2
1245      C.....+.....+.....+.....+...
1246      C
1247      DP22 = P21
1248      TSY3 = SIN(Y(3))
1249      TCY3 = COS(Y(3))
1250      TSP2 = TSY3
1251      TCP2 = TCY3
1252      DP21 = -P22
1253      AOR = AR*AR*AR
1254      RC2 = G(2,2)*TCP2+G(1,2)*TSP2
1255      BR = -(AOR+AOR)*(G(2,1)*P21+RC2*P22)
1256      BT = AOR*(G(2,1)*DP21+RC2*DP22)
1257      BP = AOR*(G(1,2)*TCP2-G(2,2)*TSP2)*P22
1258      C
1259      C.....+.....+.....+.....+...
1260      C \/ N = 3

```

```

1261 C.....+.....+.....+.....+.....+.....+...
1262 C
1263 IF (MGNMAX.LT.3) GO TO 130
1264 AOR = AOR*AR
1265 ERR = BERR*SQRT((BP/P22)**2+BR**2+BT**2)
1266 IF ((BM(3)*AOR).LT.ERR) GO TO 130
1267 TSP3 = (TSP2+TSP2)*TCP2
1268 TCP3 = (TCP2+TSP2)*(TCP2-TSP2)
1269 P31 = P21*P21-0.333333333
1270 P32 = P21*P22
1271 P33 = P22*P22
1272 DP31 = -P32-P32
1273 DP32 = P21*P21-P33
1274 DP33 = -DP31
1275 RC2 = G(3,2)*TCP2+G(1,3)*TSP2
1276 RC3 = G(3,3)*TCP3+G(2,3)*TSP3
1277 BR = BR-3.0*AOR*(G(3,1)*P31+RC2*P32+RC3*P33)
1278 BT = BT+AOR*(G(3,1)*DP31+RC2*DP32+RC3*DP33)
1279 BP = BP-AOR*((G(3,2)*TSP2-G(1,3)*TCP2)*P32+
1280 * 2.0*(G(3,3)*TSP3-G(2,3)*TCP3)*P33)
1281 C
1282 C.....+.....+.....+.....+.....+...
1283 C \/ N = 4
1284 C.....+.....+.....+.....+...
1285 C
1286 IF (MGNMAX.LT.4) GO TO 130
1287 AOR = AOR*AR
1288 IF ((BM(4)*AOR).LT.ERR) GO TO 130
1289 TSP4 = TSP2*TCP3+TCP2*TSP3
1290 TCP4 = TCP2*TCP3-TSP2*TSP3
1291 P41 = P21*P31-0.26666666*P21
1292 DP41 = P21*DP31+DP21*P31-0.26666666*DP21
1293 P42 = P21*P32-0.20000000*P22
1294 DP42 = P21*DP32+DP21*P32-0.20000000*DP22
1295 P43 = P21*P33
1296 DP43 = P21*DP33+DP21*P33
1297 P44 = P22*P33
1298 DP44 = 3.0*P43
1299 RC2 = G(4,2)*TCP2+G(1,4)*TSP2
1300 RC3 = G(4,3)*TCP3+G(2,4)*TSP3
1301 RC4 = G(4,4)*TCP4+G(3,4)*TSP4
1302 BR = BR-4.0*AOR*(G(4,1)*P41+RC2*P42+RC3*P43+RC4*P44)
1303 BT = BT+AOR*(G(4,1)*DP41+RC2*DP42+RC3*DP43+RC4*DP44)
1304 BP = BP-AOR*((G(4,2)*TSP2-G(1,4)*TCP2)*P42+
1305 * 2.0*(G(4,3)*TSP3-G(2,4)*TCP3)*P43+
1306 * 3.0*(G(4,4)*TSP4-G(3,4)*TCP4)*P44)
1307 C
1308 C.....+.....+.....+.....+...
1309 C \/ N = 5
1310 C.....+.....+.....+.....+...
1311 C
1312 IF (MGNMAX.LT.5) GO TO 130
1313 AOR = AOR*AR
1314 IF ((BM(5)*AOR).LT.ERR) GO TO 130
1315 TSP5 = (TSP3+TSP3)*TCP3
1316 TCP5 = (TCP3+TSP3)*(TCP3-TSP3)
1317 P51 = P21*P41-0.25714285*P31
1318 DP51 = P21*DP41+DP21*P41-0.25714285*DP31
1319 P52 = P21*P42-0.22857142*P32
1320 DP52 = P21*DP42+DP21*P42-0.22857142*DP32
1321 P53 = P21*P43-0.14285714*P33
1322 DP53 = P21*DP43+DP21*P43-0.14285714*DP33
1323 P54 = P21*P44
1324 DP54 = P21*DP44+DP21*P44
1325 P55 = P22*P44
1326 DP55 = 4.0*P54
1327 RC2 = G(5,2)*TCP2+G(1,5)*TSP2
1328 RC3 = G(5,3)*TCP3+G(2,5)*TSP3
1329 RC4 = G(5,4)*TCP4+G(3,5)*TSP4
1330 RC5 = G(5,5)*TCP5+G(4,5)*TSP5

```

```

1331      BR = BR-5.0*AOR*(G(5,1)*P51+RC2*P52+RC3*P53+RC4*P54+RC5*P55)
1332      BT = BT+AOR*(G(5,1)*DP51+RC2*DP52+RC3*DP53+RC4*DP54+RC5*DP55)
1333      BP = BP-AOR*((G(5,2)*TSP2-G(1,5)*TCP2)*P52+2.0*(G(5,3)*TSP3-
1334      *      G(2,5)*TCP3)*P53+3.0*(G(5,4)*TSP4-
1335      *      G(3,5)*TCP4)*P54+4.0*(G(5,5)*TSP5-
1336      *      G(4,5)*TCP5)*P55)
1337      C
1338      C.....+.....+.....+.....+.....+.....+...
1339      C \/ N = 6
1340      C.....+.....+.....+.....+.....+...
1341      C
1342      IF (MGNMAX.LT.6) GO TO 130
1343      AOR = AOR*AR
1344      IF ((BM(6)*AOR).LT.ERR) GO TO 130
1345      TSP6 = TSP2*TCP5+TCP2*TSP5
1346      TCP6 = TCP2*TCP5-TSP2*TSP5
1347      P61 = P21*P51-0.25396825*P41
1348      DP61 = P21*DP51+DP21*P51-0.25396825*DP41
1349      P62 = P21*P52-0.23809523*P42
1350      DP62 = P21*DP52+DP21*P52-0.23809523*DP42
1351      P63 = P21*P53-0.19047619*P43
1352      DP63 = P21*DP53+DP21*P53-0.19047619*DP43
1353      P64 = P21*P54-0.11111111*P44
1354      DP64 = P21*DP54+DP21*P54-0.11111111*DP44
1355      P65 = P21*P55
1356      DP65 = P21*DP55+DP21*P55
1357      P66 = P22*P55
1358      DP66 = 5.0*P65
1359      RC2 = G(6,2)*TCP2+G(1,6)*TSP2
1360      RC3 = G(6,3)*TCP3+G(2,6)*TSP3
1361      RC4 = G(6,4)*TCP4+G(3,6)*TSP4
1362      RC5 = G(6,5)*TCP5+G(4,6)*TSP5
1363      RC6 = G(6,6)*TCP6+G(5,6)*TSP6
1364      BR = BR-6.0*AOR*(G(6,1)*P61+RC2*P62+RC3*P63+RC4*P64+RC5*P65
1365      *      +RC6*P66)
1366      BT = BT+AOR*(G(6,1)*DP61+RC2*DP62+RC3*DP63+RC4*DP64+RC5*DP65
1367      *      +RC6*DP66)
1368      BP = BP-AOR*((G(6,2)*TSP2-G(1,6)*TCP2)*P62+2.0*(G(6,3)*TSP3
1369      *      -G(2,6)*TCP3)*P63+3.0*(G(6,4)*TSP4
1370      *      -G(3,6)*TCP4)*P64+4.0*(G(6,5)*TSP5
1371      *      -G(4,6)*TCP5)*P65+5.0*(G(6,6)*TSP6-G(5,6)*TCP6)*P66)
1372      C
1373      C.....+.....+.....+.....+.....+.....+...
1374      C \/ N = 7
1375      C.....+.....+.....+.....+.....+...
1376      C
1377      IF (MGNMAX.LT.7) GO TO 130
1378      AOR = AOR*AR
1379      IF ((BM(7)*AOR).LT.ERR) GO TO 130
1380      TSP7 = (TSP4+TSP4)*TCP4
1381      TCP7 = (TCP4+TSP4)*(TCP4-TSP4)
1382      P71 = P21*P61-0.25252525*P51
1383      DP71 = P21*DP61+DP21*P61-0.25252525*DP51
1384      P72 = P21*P62-0.24242424*P52
1385      DP72 = P21*DP62+DP21*P62-0.24242424*DP52
1386      P73 = P21*P63-0.21212121*P53
1387      DP73 = P21*DP63+DP21*P63-0.21212121*DP53
1388      P74 = P21*P64-0.16161616*P54
1389      DP74 = P21*DP64+DP21*P64-0.16161616*DP54
1390      P75 = P21*P65-0.09090909*P55
1391      DP75 = P21*DP65+DP21*P65-0.09090909*DP55
1392      P76 = P21*P66
1393      DP76 = P21*DP66+DP21*P66
1394      P77 = P22*P66
1395      DP77 = 6.0*P76
1396      RC2 = G(7,2)*TCP2+G(1,7)*TSP2
1397      RC3 = G(7,3)*TCP3+G(2,7)*TSP3
1398      RC4 = G(7,4)*TCP4+G(3,7)*TSP4
1399      RC5 = G(7,5)*TCP5+G(4,7)*TSP5
1400      RC6 = G(7,6)*TCP6+G(5,7)*TSP6

```

```

1401      RC7 = G(7,7)*TCP7+G(6,7)*TSP7
1402      BR = BR-7.0*AOR*(G(7,1)*P71+RC2*P72+RC3*P73+RC4*P74+RC5*P75
1403      *      +RC6*P76+RC7*P77)
1404      BT = BT+AOR*(G(7,1)*DP71+RC2*DP72+RC3*DP73+RC4*DP74+RC5*DP75
1405      *      +RC6*DP76+RC7*DP77)
1406      BP = BP-AOR*((G(7,2)*TSP2-G(1,7)*TCP2)*P72+2.0*(G(7,3)*TSP3
1407      *      -G(2,7)*TCP3)*P73+3.0*(G(7,4)*TSP4
1408      *      -G(3,7)*TCP4)*P74+4.0*(G(7,5)*TSP5
1409      *      -G(4,7)*TCP5)*P75+5.0*(G(7,6)*TSP6
1410      *      -G(5,7)*TCP6)*P76+6.0*(G(7,7)*TSP7-G(6,7)*TCP7)*P77)
1411      C
1412      C.....+.....+.....+.....+.....+.....+.....+...
1413      C \/ N = 8
1414      C.....+.....+.....+.....+.....+.....+...
1415      C
1416      IF (MGNMAX.LT.8) GO TO 130
1417      AOR = AOR*AR
1418      IF ((BM(8)*AOR).LT.ERR) GO TO 130
1419      TSP8 = TSP2*TCP7+TCP2*TSP7
1420      TCP8 = TCP2*TCP7-TSP2*TSP7
1421      P81 = P21*P71-0.25174825*P61
1422      DP81 = P21*DP71+DP21*P71-0.25174825*DP61
1423      P82 = P21*P72-0.24475524*P62
1424      DP82 = P21*DP72+DP21*P72-0.24475524*DP62
1425      P83 = P21*P73-0.22377622*P63
1426      DP83 = P21*DP73+DP21*P73-0.22377622*DP63
1427      P84 = P21*P74-0.18881118*P64
1428      DP84 = P21*DP74+DP21*P74-0.18881118*DP64
1429      P85 = P21*P75-0.13986013*P65
1430      DP85 = P21*DP75+DP21*P75-0.13986013*DP65
1431      P86 = P21*P76-0.07692307*P66
1432      DP86 = P21*DP76+DP21*P76-0.07692307*DP66
1433      P87 = P21*P77
1434      DP87 = P21*DP77+DP21*P77
1435      P88 = P22*P77
1436      DP88 = 7.0*P87
1437      RC2 = G(8,2)*TCP2+G(1,8)*TSP2
1438      RC3 = G(8,3)*TCP3+G(2,8)*TSP3
1439      RC4 = G(8,4)*TCP4+G(3,8)*TSP4
1440      RC5 = G(8,5)*TCP5+G(4,8)*TSP5
1441      RC6 = G(8,6)*TCP6+G(5,8)*TSP6
1442      RC7 = G(8,7)*TCP7+G(6,8)*TSP7
1443      RC8 = G(8,8)*TCP8+G(7,8)*TSP8
1444      BR = BR-8.0*AOR*(G(8,1)*P81+RC2*P82+RC3*P83+RC4*P84+RC5*P85
1445      *      +RC6*P86+RC7*P87+RC8*P88)
1446      BT = BT+AOR*(G(8,1)*DP81+RC2*DP82+RC3*DP83+RC4*DP84+RC5*DP85
1447      *      +RC6*DP86+RC7*DP87+RC8*DP88)
1448      BP = BP-AOR*((G(8,2)*TSP2-G(1,8)*TCP2)*P82
1449      *      +2.0*(G(8,3)*TSP3-G(2,8)*TCP3)*P83
1450      *      +3.0*(G(8,4)*TSP4-G(3,8)*TCP4)*P84
1451      *      +4.0*(G(8,5)*TSP5-G(4,8)*TCP5)*P85
1452      *      +5.0*(G(8,6)*TSP6-G(5,8)*TCP6)*P86
1453      *      +6.0*(G(8,7)*TSP7-G(6,8)*TCP7)*P87
1454      *      +7.0*(G(8,8)*TSP8-G(7,8)*TCP8)*P88)
1455      C
1456      C.....+.....+.....+.....+.....+.....+...
1457      C \/ N = 9
1458      C.....+.....+.....+.....+.....+...
1459      C
1460      IF (MGNMAX.LT.9) GO TO 130
1461      AOR = AOR*AR
1462      IF ((BM(9)*AOR).LT.ERR) GO TO 130
1463      TSP9 = (TSP5+TSP5)*TCP5
1464      TCP9 = (TCP5+TSP5)*(TCP5-TSP5)
1465      P91 = P21*P81-0.25128205*P71
1466      DP91 = P21*DP81+DP21*P81-0.25128205*DP71
1467      P92 = P21*P82-0.24615384*P72
1468      DP92 = P21*DP82+DP21*P82-0.24615384*DP72
1469      P93 = P21*P83-0.23076923*P73
1470      DP93 = P21*DP83+DP21*P83-0.23076923*DP73

```

```

1471      P94 = P21*P84-0.20512820*P74
1472      DP94 = P21*DP84+DP21*P84-0.20512820*DP74
1473      P95 = P21*P85-0.16923076*P75
1474      DP95 = P21*DP85+DP21*P85-0.16923076*DP75
1475      P96 = P21*P86-0.12307692*P76
1476      DP96 = P21*DP86+DP21*P86-0.12307692*DP76
1477      P97 = P21*P87-0.06666666*P77
1478      DP97 = P21*DP87+DP21*P87-0.06666666*DP77
1479      P98 = P21*P88
1480      DP98 = P21*DP88+DP21*P88
1481      P99 = P22*P88
1482      DP99 = 8.0*P98
1483      RC2 = G(9,2)*TCP2+G(1,9)*TSP2
1484      RC3 = G(9,3)*TCP3+G(2,9)*TSP3
1485      RC4 = G(9,4)*TCP4+G(3,9)*TSP4
1486      RC5 = G(9,5)*TCP5+G(4,9)*TSP5
1487      RC6 = G(9,6)*TCP6+G(5,9)*TSP6
1488      RC7 = G(9,7)*TCP7+G(6,9)*TSP7
1489      RC8 = G(9,8)*TCP8+G(7,9)*TSP8
1490      RC9 = G(9,9)*TCP9+G(8,9)*TSP9
1491      BR = BR-9.0*AOR*(G(9,1)*P91+RC2*P92+RC3*P93+RC4*P94+RC5*P95
1492      * +RC6*P96+RC7*P97+RC8*P98+RC9*P99)
1493      BT = BT+AOR*(G(9,1)*DP91+RC2*DP92+RC3*DP93+RC4*DP94+RC5*DP95
1494      * +RC6*DP96+RC7*DP97+RC8*DP98+RC9*DP99)
1495      BP = BP-AOR*((G(9,2)*TSP2-G(1,9)*TCP2)*P92+2.0*(G(9,3)*TSP3
1496      * -G(2,9)*TCP3)*P93+3.0*(G(9,4)*TSP4
1497      * -G(3,9)*TCP4)*P94+4.0*(G(9,5)*TSP5
1498      * -G(4,9)*TCP5)*P95+5.0*(G(9,6)*TSP6
1499      * -G(5,9)*TCP6)*P96+6.0*(G(9,7)*TSP7
1500      * -G(6,9)*TCP7)*P97+7.0*(G(9,8)*TSP8
1501      * -G(7,9)*TCP8)*P98+8.0*(G(9,9)*TSP9-G(8,9)*TCP9)*P99)
1502      C
1503      C.....+.....+.....+.....+.....+.....+.....+...
1504      C \/ N = 10
1505      C.....+.....+.....+.....+.....+.....+...
1506      C
1507      IF (MGNMAX.LT.10) GO TO 130
1508      AOR = AOR*AR
1509      IF ((BM(10)*AOR).LT.ERR) GO TO 130
1510      TSP10 = TSP2*TCP9+TCP2*TSP9
1511      TCP10 = TCP2*TCP9-TSP2*TSP9
1512      P101 = P21*P91-0.25098039*P81
1513      DP101 = P21*DP91+DP21*P91-0.25098039*DP81
1514      P102 = P21*P92-0.24705882*P82
1515      DP102 = P21*DP92+DP21*P92-0.24705882*DP82
1516      P103 = P21*P93-0.23529411*P83
1517      DP103 = P21*DP93+DP21*P93-0.23529411*DP83
1518      P104 = P21*P94-0.21568627*P84
1519      DP104 = P21*DP94+DP21*P94-0.21568627*DP84
1520      P105 = P21*P95-0.18823529*P85
1521      DP105 = P21*DP95+DP21*P95-0.18823529*DP85
1522      P106 = P21*P96-0.15294117*P86
1523      DP106 = P21*DP96+DP21*P96-0.15294117*DP86
1524      P107 = P21*P97-0.10980392*P87
1525      DP107 = P21*DP97+DP21*P97-0.10980392*DP87
1526      P108 = P21*P98-0.05882352*P88
1527      DP108 = P21*DP98+DP21*P98-0.05882352*DP88
1528      P109 = P21*P99
1529      DP109 = P21*DP99+DP21*P99
1530      P1010 = P22*P99
1531      DP1010 = 9.0*P109
1532      RC2 = G(10,2)*TCP2+G(1,10)*TSP2
1533      RC3 = G(10,3)*TCP3+G(2,10)*TSP3
1534      RC4 = G(10,4)*TCP4+G(3,10)*TSP4
1535      RC5 = G(10,5)*TCP5+G(4,10)*TSP5
1536      RC6 = G(10,6)*TCP6+G(5,10)*TSP6
1537      RC7 = G(10,7)*TCP7+G(6,10)*TSP7
1538      RC8 = G(10,8)*TCP8+G(7,10)*TSP8
1539      RC9 = G(10,9)*TCP9+G(8,10)*TSP9
1540      RC10 = G(10,10)*TCP10+G(9,10)*TSP10

```

```

1541      BR = BR-10.0*AOR*(G(10,1)*P101+RC2*P102+RC3*P103+RC4*P104
1542      * +RC5*P105+RC6*P106+RC7*P107+RC8*P108+RC9*P109+RC10*P1010)
1543      BT = BT+AOR*(G(10,1)*DP101+RC2*DP102+RC3*DP103+RC4*DP104
1544      * +RC5*DP105+RC6*DP106+RC7*DP107+RC8*DP108+RC9*DP109
1545      * +RC10*DP1010)
1546      BP = BP-AOR*((G(10,2)*TSP2-G(1,10)*TCP2)*P102+2.0*(G(10,3)*TSP3
1547      * -G(2,10)*TCP3)*P103+3.0*(G(10,4)*TSP4
1548      * -G(3,10)*TCP4)*P104+4.0*(G(10,5)*TSP5
1549      * -G(4,10)*TCP5)*P105+5.0*(G(10,6)*TSP6
1550      * -G(5,10)*TCP6)*P106+6.0*(G(10,7)*TSP7
1551      * -G(6,10)*TCP7)*P107+7.0*(G(10,8)*TSP8
1552      * -G(7,10)*TCP8)*P108+8.0*(G(10,9)*TSP9
1553      * -G(8,10)*TCP9)*P109+9.0*(G(10,10)*TSP10
1554      * -G(9,10)*TCP10)*P1010)
1555      C
1556      C.....+.....+.....+.....+.....+.....+...
1557      C \/ N = 11
1558      C.....+.....+.....+.....+.....+...
1559      C
1560      IF (MGNMAX.LT.11) GO TO 130
1561      AOR = AOR*AR
1562      IF ((BM(11)*AOR).LT.ERR) GO TO 130
1563      TSP11 = (TSP6+TSP6)*TCP6
1564      TCP11 = (TCP6+TSP6)*(TCP6-TSP6)
1565      P111 = P21*P101-0.25077399*P91
1566      DP111 = P21*DP101+DP21*P101-0.25077399*DP91
1567      P112 = P21*P102-0.24767801*P92
1568      DP112 = P21*DP102+DP21*P102-0.24767801*DP92
1569      P113 = P21*P103-0.23839009*P93
1570      DP113 = P21*DP103+DP21*P103-0.23839009*DP93
1571      P114 = P21*P104-0.22291021*P94
1572      DP114 = P21*DP104+DP21*P104-0.22291021*DP94
1573      P115 = P21*P105-0.20123839*P95
1574      DP115 = P21*DP105+DP21*P105-0.20123839*DP95
1575      P116 = P21*P106-0.17337461*P96
1576      DP116 = P21*DP106+DP21*P106-0.17337461*DP96
1577      P117 = P21*P107-0.13931888*P97
1578      DP117 = P21*DP107+DP21*P107-0.13931888*DP97
1579      P118 = P21*P108-0.09907120*P98
1580      DP118 = P21*DP108+DP21*P108-0.09907120*DP98
1581      P119 = P21*P109-0.05263157*P99
1582      DP119 = P21*DP109+DP21*P109-0.05263157*DP99
1583      P1110 = P21*P1010
1584      DP1110 = P21*DP1010+DP21*P1010
1585      P1111 = P22*P1010
1586      DP1111 = 10.0*P1110
1587      RC2 = G(11,2)*TCP2+G(1,11)*TSP2
1588      RC3 = G(11,3)*TCP3+G(2,11)*TSP3
1589      RC4 = G(11,4)*TCP4+G(3,11)*TSP4
1590      RC5 = G(11,5)*TCP5+G(4,11)*TSP5
1591      RC6 = G(11,6)*TCP6+G(5,11)*TSP6
1592      RC7 = G(11,7)*TCP7+G(6,11)*TSP7
1593      RC8 = G(11,8)*TCP8+G(7,11)*TSP8
1594      RC9 = G(11,9)*TCP9+G(8,11)*TSP9
1595      RC10 = G(11,10)*TCP10+G(9,11)*TSP10
1596      RC11 = G(11,11)*TCP11+G(10,11)*TSP11
1597      BR = BR-11.0*AOR*(G(11,1)*P111+RC2*P112+RC3*P113+RC4*P114
1598      * +RC5*P115+RC6*P116+RC7*P117+RC8*P118+RC9*P119+RC10*P1110
1599      * +RC11*P1111)
1600      BT = BT+AOR*(G(11,1)*DP111+RC2*DP112+RC3*DP113+RC4*DP114
1601      * +RC5*DP115+RC6*DP116+RC7*DP117+RC8*DP118+RC9*DP119
1602      * +RC10*DP1110+RC11*DP1111)
1603      BP = BP-AOR*((G(11,2)*TSP2-G(1,11)*TCP2)*P112+2.0*(G(11,3)*TSP3
1604      * -G(2,11)*TCP3)*P113 + 3.0 *(G(11,4)*TSP4
1605      * -G(3,11)*TCP4)*P114 + 4.0 *(G(11,5)*TSP5
1606      * -G(4,11)*TCP5)*P115 + 5.0 *(G(11,6)*TSP6
1607      * -G(5,11)*TCP6)*P116 + 6.0 *(G(11,7)*TSP7
1608      * -G(6,11)*TCP7)*P117 + 7.0 *(G(11,8)*TSP8
1609      * -G(7,11)*TCP8)*P118 + 8.0 *(G(11,9)*TSP9
1610      * -G(8,11)*TCP9)*P119 + 9.0 *(G(11,10)*TSP10

```

```
1611      *      -G(9,11)*TCP10)*P1110+10.0*(G(11,11)*TSP11
1612      *      -G(10,11)*TCP11)*P1111)
1613  C
1614  C.....+.....+.....+.....+.....+.....+...
1615  C    \/\ Convert to units of Gauss
1616  C.....+.....+.....+.....+.....+...
1617  C
1618  130 BP = BP/P22*1.E-5
1619  BT = BT*1.E-5
1620  BR = BR*1.E-5
1621  B = SQRT(BR*BR+BT*BT+BP*BP)
1622  RETURN
1623  C
1624  C.....+.....+.....+.....+.....+.....+...
1625  C    Y(1) is R coordinate      Y(2) is THETA coordinate
1626  C    Y(3) is PHI coordinate    Y(4) is V(R)
1627  C    Y(5) is V(THETA)        Y(6) is V(PHI)
1628  C    F(1) is R dot          F(2) is THETA dot
1629  C    F(3) is PHI dot        F(4) is R dot dot
1630  C    F(5) is THETA dot dot  F(6) is PHI dot dot
1631  C    BR   is B(R)          BT   is B(THETA)
1632  C    BP   is B(PHI)        B    is magnitude of magnetic field
1633  C.....+.....+.....+.....+.....+...
1634  C
1635  END
```

```

1      PROGRAM TJI95T
2      C
3      C.....+.....+.....+.....+.....+.....+...
4      C Multi-platform COSMIC-RAY TRAJECTORY PROGRAM
5      C FORTRAN 77 transportable version
6      C Read in control card; LAT, LON, RIG, ZENITH, AZIMUTH, DELPC, INDO
7      C     Then calculate INDO trajectories
8      C     Starting at PC
9      C     Incrementing at DELPC intervals
10     C     Includes conversion from Geodetic to Geocentric coordinates
11     C     Includes re-entrant albedo calculations
12     C     Uses subroutine SINGLTJE to do trajectory calculations
13     C     Magnetic field - IGRF 1995 (order 10)      ###
14     C.....+.....+.....+.....+.....+.....+...
15     C     Restrictions: Cannot run over N or S pole; will get BETA blowup
16     C.....+.....+.....+.....+.....+.....+...
17     C     Mod History
18     CLast Mod 21 Dec 00 Make all intrinsic function double precision for PC
19     C Mod 20 Dec 00 Insert 8 character format 1000 with AZ & ZE
20     C Mod 17 Feb 99 set limit to 600000
21     C Mod 17 Feb 99 if (ymax.lt.6.6) IFATE = 3
22     C Mod Aug 97 Adjust step size to minimize beta problems
23     C Mod Jan 97 High latitude step size adjust, introduce AHLT
24     C Mod Jun 96 EDIF limit set to 1.0e-5
25     C Mod Jun 96 IERRPT formats, Boundary and look ahead
26     C Mod Feb 96 Standard reference TJ1V line check
27     C Mod Dec 94 Print out start and end times of PC run
28     C ****
29     C     Timing estimates base on COMPAQ Digital FORTRAN
30     C Will run on PIII PC at 850 MHZ      55000 steps/sec (Real*8)
31     C Will run on PIII PC at 700 MHZ      39000 steps/sec (Real*8)
32     C Will run on PIII PC at 550 MHZ      32000 steps/sec (Real*8)
33     C Will run on PII PC at 400 MHZ      23000 steps/sec (Real*8)
34     C ****
35     C     * TAPE*      Monitor program operation
36     C     * TAPE1      Trajectory control cards
37     C     * TAPE7      80 character line (card image) output
38     C     * TAPE8      132 character line printer output
39     C     * TAPE16     Diagnostic output for trouble shooting
40     C     *          Normally turned off (open statement commented out)
41     C ****
42     C.....+.....+.....+.....+.....+...
43     C Programmer - Don F. Smart; FORTRAN77
44     C Note - The programming adheres to the conventional FORTRAN
45     C default standard that variables beginning with
46     C 'i','j','k','l','m',or 'n' are integer variables
47     C Variables beginning with "c" are character variables
48     C All other variables are real
49     C.....+.....+.....+.....+.....+...
50     C Do not mix different type variables in same common block
51     C Some computers do not allow this
52     C.....+.....+.....+.....+.....+...
53     C
54     IMPLICIT INTEGER (I-N)
55     IMPLICIT REAL * 8(A-B)
56     IMPLICIT REAL * 8(D-H)
57     IMPLICIT REAL * 8(O-Z)
58     C
59     C.....+.....+.....+.....+.....+...
60     C \/ The following used for timing PC runs
61     C Can use on PC (IBM and COMPACQ FORTRAN) and on SUN's
62     C Cannot use on IBM SP2 or DEC VAX
63     C
64     C.....+.....+.....+.....+...
65     INTEGER*2 ISYEAR,ISMONT,ISDAY, ISHOUR,ISMIN,ISSEC,ISHSEC
66     INTEGER*2 IEYEAR,IEMONTH,IEDAY, IEHOUR,IEMIN,IESEC,IEHSEC
67     C
68     C.....+.....+.....+.....+.....+...
69     C
70     COMMON /WRKVLU/ F(6),Y(6),ERAD,EOMC,VEL,BR,BT,BP,B

```

```

71      COMMON /WRKTSC/ TSY2,TCY2,TSY3,TCY3
72      COMMON /TRIG/    PI,RAD,PIO2
73      COMMON /GEOID/   ERADPL, ERECSQ
74      COMMON /SNGLR/   SALT,DISOUT,GCLATD,GDLATD,GLOND,GAZD,GDZED,
75      *              RY1,RY2,RY3,RHT,TSTEP
76      COMMON /SNGLI/   LIMIT,NTRAJC,IERRPT
77      C
78      C.....+.....+.....+.....+.....+.....+.....+...
79      C
80      CHARACTER*1 CFF
81      C
82      C.....+.....+.....+.....+.....+.....+.....+...
83      C  \/ Use CFF / '1' / on 'Main Fnames';
84      C      use CFF /Z'0C' / for files to be printed on word processors
85      C.....+.....+.....+.....+.....+.....+...
86      C
87      DATA CFF / '1'
88      DATA CFF /Z'0C'
89      C
90      C.....+.....+.....+.....+.....+...
91      C
92      OPEN (1, FILE='TAPE1', STATUS='OLD')
93      OPEN (7, FILE='TAPE7', STATUS='UNKNOWN')
94      OPEN (8, FILE='TAPE8', STATUS='UNKNOWN')
95      open (16,FILE='TAPE16',STATUS='UNKNOWN')
96      C
97      C.....+.....+.....+.....+.....+.....+...
98      C  \/ Get date and time of run start (works on PC's)
99      C.....+.....+.....+.....+.....+...
100     C
101     CALL GETDAT (ISYEAR,ISMONT,ISDAY)
102     CALL GETTIM (ISHOUR,ISMIN,ISSEC,ISHSEC)
103     C
104     WRITE (8, 1000) CFF, ISYEAR,ISMONT, ISDAY,ISHOUR,ISMIN,ISSEC
105     WRITE (16,1000) CFF, ISYEAR,ISMONT, ISDAY,ISHOUR,ISMIN,ISSEC
106     1000 FORMAT (A1,' RUN START DATE ', I4, '/',I2, '/',I2,'@',I2,':',I2,
107     *           ::',I2)
108     C
109     C.....+.....+.....+.....+.....+...
110     C  \/ User defined program control
111     C.....+.....+.....+...
112
113     FSTEP = 4.0E08
114     LIMIT = 600000
115     C
116     C.....+.....+.....+.....+.....+.....+...
117     C  \/ FSTEP is total number of steps before run is terminated
118     C  LIMIT is max number of steps before trajectory declared F
119     C.....+.....+.....+.....+.....+...
120     C  \/ Define program constants
121     C.....+.....+.....+.....+.....+...
122     C  DISOUT is radial distance for trajectory termination
123     C  ERAD is average earth radius
124     C  NTRAJC is number of trajectory computed in this run
125     C  RHT is top of atmosphere for re-entrant trajectory
126     C  TSTEP is number of steps executed in this run
127     C.....+.....+.....+.....+.....+...
128     C
129     NTRAJC = 0
130     TSTEP = 0.0
131     C
132     DISOUT = 25.0
133     ERAD = 6371.2
134     RHT = 20.0
135     VEL = 2.99792458E5/ERAD
136     C
137     C.....+.....+.....+.....+.....+...
138     C  "VEL" is light velocity in earth radii per second
139     C  Light speed defined as 299792458 m/s
140     C  Ref: E. R. Cohn AND B. N. Taylor, "The Fundamental Physical

```

```

141 C           Constants, Physics Today Pl1, August 1987.
142 C.....+.....+.....+.....+.....+.....+.....+...
143 C   \/ Define essential trigonometric values
144 C.....+.....+.....+.....+.....+.....+...
145 C
146     PI    = ACOS(-1.0)
147     RAD   = 180.0/PI
148     PIO2  = PI/2.0
149 C
150 C.....+.....+.....+.....+.....+.....+...
151 C   \/ TAPE1 must contain trajectory control cards
152 C       Terminate program if no data on TAPE1 file
153 C       Terminate if EOF encountered
154 C       Terminate if negative data found on input file
155 C       Terminate if bad      data found on input file
156 C.....+.....+.....+.....+.....+...
157 C
158   100 READ (1,1010,IOSTAT=IOSTAT,ERR=120,END=110) GDLATD,GLOND,PC,
159     *          GDZED,GDAZD,DELPC,INDO,IERRPT,INDEX
160   1010 FORMAT (BZ,6F8.2,3I8)
161 C
162     110 CONTINUE
163     IF (IOSTAT.LT.0) THEN
164       WRITE (*,1020)
165       GO TO 150
166     ENDIF
167   1020 FORMAT (' END OF FILE ON TAPE 1 (DATA INPUT)')
168
169   120 IF (IOSTAT.GT.0) THEN
170     WRITE (*,1030) IOSTAT,GDLATD,GLOND,PC,DELPC,
171     *                  INDO,IERRPT,INDEX
172     GO TO 150
173   ENDIF
174   1030 FORMAT (' ERROR ON DATA INPUT FILE (TAPE1), IOSTAT =',I5/
175     *        4F8.3,3I8)
176 C
177     IF (PC.LE.0) THEN
178       WRITE (*,1040)
179       GO TO 150
180     ENDIF
181   1040 FORMAT (' END OF DATA INPUT (NEGATIVE VALUE READ IN)')
182 C
183     WRITE (*,1050) GDLATD,GLOND,PC,GDZED,GDAZD,DELPC,INDO,IERRPT,INDEX
184   1050 FORMAT (' TAPE 1 ',6F7.2,3I6)
185 C
186 C.....+.....+.....+.....+.....+.....+...
187 C   \/ Start at top of atmosphere (20 km above surface of oblate earth)
188 C       Coding is relic of past when ISALT was read in
189 C.....+.....+.....+.....+.....+...
190 C
191     ISALT = 0
192     IF (ISALT.LE.0) SALT = 20.0
193     IF (ISALT.GT.0) SALT = ISALT
194 C
195     KNT = 0
196     IDELPC = DELPC*1000.0+0.0001
197     INDXPC = PC*1000.0+0.0001
198 C
199 C.....+.....+.....+.....+.....+...
200 C   For trajectories from Earth
201 C       convert from Geodetic coordinates to Geocentric coordinates
202 C       Geodetic   coordinates used for input
203 C       Geocentric coordinates used for output
204 C       All calculation are done in Geocentric coordinates!
205 C   \/ Conversion from Geodetic to Geocentric coordinates
206 C.....+.....+.....+.....+...
207 C
208     CALL GDGC (TCD, TSD)
209 C
210 C.....+.....+.....+.....+.....+...

```

```

211 C      /* Remember positron of initial point on trajectory
212 C          in Geocentric coordinates
213 C      Y(1) is distance in earth radii from geocenter
214 C          Start with height above geoid and convert to earth radii
215 C          The initial values of Y(1), Y(2), and Y(3) are
216 C          calculated in subroutine GDGC
217 C          Coordinate reference system
218 C              Y(1) = R      = vertical
219 C              Y(2) = THETA = south
220 C              Y(3) = PHI   = east
221 C.....+.....+.....+.....+.....+.....+...
222 C
223     RY2 = Y(2)
224     RY3 = Y(3)
225     RY1 = Y(1)
226 C
227     GDAZ = GDAZD/RAD
228     GDZE = GDZED/RAD
229     TSGDZE = SIN(GDZE)
230     TCGDZE = COS(GDZE)
231     TSGDAZ = SIN(GDAZ)
232     TCGDAZ = COS(GDAZ)
233 C
234 C.....+.....+.....+.....+.....+.....+...
235 C      /* Get Y1, Y2, Y3 components in Geodetic coordinates
236 C          Azimuth is measured clockwise from the north
237 C          in R, THETA, PHI coordinates, in the THETA-PHI plane
238 C          The angle is 180 - AZD
239 C.....+.....+.....+.....+.....+...
240 C
241     Y1GD = TCGDZE
242     Y2GD = -TSGDZE*TCGDAZ
243     Y3GD = TSGDZE*TSGDAZ
244 C
245 C.....+.....+.....+.....+.....+.....+...
246 C      /* The small angle delta at the point in space between the
247 C          downward Geodetic direction and the
248 C          downward Geocentric direction is given by
249 C          DELTA = Geocentric co-latitude + Geodetic latitude - 90 (deg)
250 C
251 C          We are looking up
252 C          The rotation from Geodetic vertical to Geocentric Vertical
253 C          Is always rotation toward the equator
254 C
255 C      /* Convert from Geodetic to Geocentric Components for Y1, Y2,
256 C.....+.....+.....+.....+.....+...
257 C
258     Y1GC = Y1GD*TCD+Y2GD*TSD
259     Y2GC = -Y1GD*TSD+Y2GD*TCD
260     Y3GC = Y3GD
261 C
262 C      WRITE (*,1060) GDZED, GDZE, GDAZD, GDAZ, TSGDZE, TCGDZE, TSGDAZ, TCGDAZ
263 C      WRITE (*,1060) Y1GD, Y2GD, Y3GD, Y1GC, Y2GC, Y3GC
264 C1060 FORMAT (' 1050',8F15.5)
265 C
266 C.....+.....+.....+.....+.....+.....+...
267 C      ****
268 C      Main control of trajectory calculations begins here
269 C      Trajectories are calculated in subroutine SINGLTJ
270 C      ****
271 C
272 C      PC      = rigidity IN GV
273 C      INDXPC = index of rigidity in MV (integer)
274 C      IRSLT  = trajectory result
275 C          IRSLT +1    allowed
276 C          IRSLT 0    failed
277 C          IRSLT -1   re-entrant
278 C.....+.....+.....+.....+.....+...
279 C
280     DO 130 NDO = 1, INDO

```

```

281 C
282     IF (IERRPT.GE.1) WRITE (16,1070) GDLATD,GLOND,KNT,INDO,NDO,
283     * IDELPC,INDXPC,DELPC, PC
284 C
285     CALL SINGLTJ (PC,IRSLT,INDXPC,Y1GC,Y2GC,Y3GC)
286 C
287     KNT = KNT+1
288     INDXPC = INDXPC-IDE LPC
289     PC = FLOAT(INDXPC)/1000.0
290 C
291 C      +.....+.....+.....+.....+.....+.....+...
292 C      \/ Check termination conditions
293 C      +.....+.....+.....+.....+.....+...
294 C
295     IF (PC .LE. 0.0) GO TO 140
296     IF (TSTEP .GE. FSTEP) GO TO 150
297 C
298     130 CONTINUE
299     140 CONTINUE
300    1070 FORMAT (' 1070 ',2F7.2,5I6,2F6.2)
301 C
302 C      +.....+.....+.....+.....+.....+.....+...
303 C      ****
304 C      End of main control loop
305 C      ****
306 C      \/ Go read in next control card
307 C      +.....+.....+.....+.....+.....+...
308 C
309     GO TO 100
310 C
311 C      +.....+.....+.....+.....+.....+...
312 C      ****
313 C      End of trajectory calculations
314 C      ****
315 C      +.....+.....+.....+.....+.....+...
316 C
317     150 CONTINUE
318 C
319 C      +.....+.....+.....+.....+.....+...
320 C      \/ Get date and time of run end (PC routine)
321 C      +.....+.....+.....+.....+.....+...
322 C
323     CALL GETDAT (IEYEAR, IEMONTH, IEDAY)
324     CALL GETTIM (IEHOUR, IEMIN, IESEC, IEHSEC)
325 C
326     WRITE (8, 1100) IEYEAR, IEMONTH, IEDAY, IEHOUR, IEMIN, IESEC
327     WRITE (16,1100) IEYEAR, IEMONTH, IEDAY, IEHOUR, IEMIN, IESEC
328     1100 FORMAT ('// RUN END DATE ', I4, '/',I2, '/',I2,'@',I2,':',I2,
329     *           ':',I2)
330     WRITE (8, 1110) ISYEAR,ISMONT, ISDAY,ISHOUR,ISMIN,ISSEC
331     1110 FORMAT (' RUN START DATE ', I4, '/',I2, '/',I2,'@',I2,':',I2,
332     *           ':',I2)
333 C
334     WRITE (*, 1120) TSTEP,NTRAJC
335     WRITE (8, 1120) TSTEP,NTRAJC
336     WRITE (16,1120) TSTEP,NTRAJC
337     1120 FORMAT ('// TOTAL NUMBER OF STEPS      ',F15.0//)
338     *          ' TOTAL NUMBER OF TRAJECTORIES',I15//)
339     Write (*,1130)
340     1130 format (' End program TJI95T')
341 C
342     STOP
343 C
344 C      +.....+.....+.....+.....+.....+...
345 C      Y(1) is R coordinate      Y(2) is THETA coordinate
346 C      Y(3) is PHI coordinate     Y(4) is V(R)
347 C      Y(5) is V(THETA)          Y(6) is V(PHI)
348 C      F(1) is R dot            F(2) is THETA dot
349 C      F(3) is PHI dot          F(4) is R dot dot
350 C      F(5) is THETA dot dot   F(6) is PHI dot dot

```

```

351 C     BR   is B(R)           BT   is B(THETA)
352 C     BP   is B(PHI)         B    is magnitude of magnetic field
353 C.....+.....+.....+.....+.....+.....+...
354 C
355 C     ierrpt vlu Program Format Variables printed out
356 C     IERRPT = 1 "MAIN" 1070 Input to SINGLTJ
357 C     IERRPT = 1 SINGLTJ 2000 Input to SINGLTJ
358 C     IERRPT = 2 SINGLTJ 2070 PC,BETA,KBF,RCKBETA,NSTEP,TBETA,Y,H
359 C     IERRPT = 4 SINGLTJ 2090 Y,F,ACCR,H,NSTEP
360 C     IERRPT = 3 SINGLTJ 2100 H,HCK,Y(1),DELACC,PC,NSTEP
361 C     IERRPT = 3 SINGLTJ 2110 H,HCK,Y(1),RFA,    PC,NSTEP
362 C     IERRPT = 3 SINGLTJ 2120 H,HCK,Y(1),NAMX,F(ICK),ICK,FOLD(ICK),
363 C                           ICK,PC,STEP
364 C     IERRPT = 4 SINGLTJ 2130 Y(1),DISCK,PVEL,H,HSNEK,HOLD,NSTEP
365 C     IERRPT = 4 SINGLTJ 2140 Y(1),DISCK,PVEL,H,      HOLD,NSTEP
366 C
367 C     END
368 C     SUBROUTINE GDGC (TCD, TSD)
369 C
370 C.....+.....+.....+.....+.....+.....+...
371 C     /* Convert from Geodetic to Geocentric coordinates
372 C     Adopted from NASA ALLMAG
373 C     GDLATD = Geodetic latitude (in degrees)
374 C     GCLATD = Geocentric latitude (in degrees)
375 C     GDCLT = Geodetic co-latitude
376 C     ERPLSQ is earth radius AT poles squared = 40408585 (km sq)
377 C     EREQSQ is earth radius AT equator squared = 40680925 (km sq)
378 C     ERADPR is earth polar radius = 6356.774733 (km)
379 C     ERADER is earth equatorial radius = 6378.160001 (km)
380 C     ERAD is earth average radius = 6371.25 (km)
381 C     ERADFL is flattening factor = 1.0/298.25
382 C     ERADFL = (ERADEQ - factor)/ERADEQ
383 C     ERECSQ is eccentricity squared = 0.00673966
384 C     ERECSQ = EREQSQ/ERPLSQ - 1.0
385 C.....+.....+.....+.....+...
386 C
387 C     CLast Mod 15 Jan 97 Common block SNGLR & SNGLI
388 C     Mod Feb 96 Standard reference TJ1V line check
389 C
390 C.....+.....+.....+.....+...
391 C     Programmer - Don F. Smart; FORTRAN77
392 C     Note - The programming adheres to the conventional FORTRAN
393 C             default standard that variables beginning with
394 C             'i','j','k','l','m', or 'n' are integer variables
395 C             Variables beginning with "c" are character variables
396 C             All other variables are real
397 C.....+.....+.....+.....+...
398 C             Do not mix different type variables in same common block
399 C             Some computers do not allow this
400 C.....+.....+.....+...
401 C
402 C     IMPLICIT INTEGER (I-N)
403 C     IMPLICIT REAL * 8(A-B)
404 C     IMPLICIT REAL * 8(D-H)
405 C     IMPLICIT REAL * 8(O-Z)
406 C
407 C.....+.....+.....+...
408 C
409 C     COMMON /WRKVLU/ F(6),Y(6),ERAD,EOMC,VEL,BR,BT,BP,B
410 C     COMMON /WRKTSC/ TSY2,TCY2,TSY3,TCY3
411 C     COMMON /TRIG/ PI,RAD,PIO2
412 C     COMMON /GEOID/ ERADEP, ERECSQ
413 C     COMMON /SNGLR/ SALT,DISOUT,GCLATD,GDLATD,GLOND,GAZD,GDZED,
414 C                      RY1,RY2,RY3,RHT,TSTEP
415 C
416 C.....+.....+...
417 C
418 C     ERPLSQ = 40408585.0
419 C     EREQSQ = 40680925.0
420 C     ERADFL = SQRT(ERPLSQ)

```

```

421      ERECSQ = EREQSQ/ERPLSQ - 1.0
422      C
423      GDCLT = PIO2-GDLATD/RAD
424      TSGDCLT = SIN(GDCLT)
425      TCGDCLT = COS(GDCLT)
426      ONE = EREQSQ*TSGDCLT*TSGDCLT
427      TWO = ERPLSQ*TCGDCLT*TCGDCLT
428      THREE = ONE+TWO
429      RHO = SQRT(THREE)
430      C
431      C.....+.....+.....+.....+.....+.....+...
432      C     \/ Get geocentric distance from geocenter in kilometers
433      C.....+.....+.....+.....+.....+...
434      C
435      DISTKM = SQRT(SALT* (SALT+2.0*RHO)+(EREQSQ*ONE+ERPLSQ*TWO) /THREE)
436      C
437      C.....+.....+.....+.....+.....+.....+...
438      C     TCD and TSD are sine and cosine of the angle the Geodetic vertical
439      C           must be rotated to form the Geocentric vertical
440      C.....+.....+.....+.....+.....+...
441      C
442      TCD = (SALT+RHO)/DISTKM
443      TSD = (EREQSQ-ERPLSQ)/RHO*TCGDCLT*TSGDCLT/DISTKM
444      TCY2 = TCGDCLT*TCD-TSGDCLT*TSD
445      TSY2 = TSGDCLT*TCD+TCGDCLT*TSD
446      C
447      Y(2) = ACOS(TCY2)
448      Y(3) = GLOND/RAD
449      Y(1) = DISTKM/ERAD
450      C
451      GCLATD = (PIO2-Y(2))*RAD
452      C
453      C     WRITE (*,1200) GDLATD,GDCLT,TSGDCLT,TCGDCLT,ONE,TWO,THREE,RHO
454      C1200 FORMAT (' 1200',8F15.5)
455      C     WRITE (*,1200) DISTKM,TCD,TSD,TCY2,TSY2,GCLATD
456      C
457      RETURN
458      END
459      SUBROUTINE SINGLTJ (PC,IRSLT,INDXPC,Y1GC,Y2GC,Y3GC)
460      C
461      C.....+.....+.....+.....+.....+.....+...
462      C     Cosmic-ray trajectory calculations subroutine
463      C           calculates cosmic ray trajectory at rigidity PC
464      C.....+.....+.....+.....+.....+...
465      C     PC      = rigidity in GV
466      C     IRSLT   = trajectory result
467      C     INDXPC = index of rigidity in mv (integer)
468      C           Y1GC,Y2GC,Y3GC are initial geocentric coordinates
469      C.....+.....+.....+.....+.....+...
470      C     \/ Step size optimization & look ahead for potential BETA problems
471      C           monitor accelerating terms and reduce step length
472      C           if large increase occurs
473      C           Restart at smaller step size if BETA error occurs
474      C.....+.....+.....+.....+.....+...
475      C     Restrictions: Cannot run over N or S pole; will get BETA blowup
476      C.....+.....+.....+.....+...
477      CLast Mod 17 Feb 99  if (ymax.lt.6.6) IFATE = 3
478      C     Mod 18 Jan 97  Patch high latitude beta problem
479      C     Mod  Jan 97  High latitude step size adjust, introduce AHLT
480      C     Mod  Jun 96  EDIF limit set to 1.0e-5
481      C     Mod  Jun 96  IERRPT formats, Boundary and look ahead
482      C     Mod  FEB 96  standard reference TJ1V (line check 17 Feb)
483
484      C.....+.....+.....+.....+.....+...
485      C     Programmer - Don F. Smart; FORTRAN77
486      C     Note - The programming adheres to the conventional FORTRAN
487      C           default standard that variables beginning with
488      C           'i','j','k','l','m',or 'n' are integer variables
489      C           Variables beginning with "c" are character variables
490      C           All other variables are real

```

```
491 C.....+.....+.....+.....+.....+.....+...
492 C      Do not mix different type variables in same common block
493 C      Some computers do not allow this
494 C.....+.....+.....+.....+.....+...
495 C
496      IMPLICIT INTEGER (I-N)
497      IMPLICIT REAL * 8(A-B)
498      IMPLICIT REAL * 8(D-H)
499      IMPLICIT REAL * 8(O-Z)
500 C
501 C.....+.....+.....+.....+.....+...
502 C
503      COMMON /WRKVLU/ F(6),Y(6),ERAD,EOMC,VEL,BR,BT,BP,B
504      COMMON /WRKTSC/ TSY2,TCY2,TSY3,TCY3
505      COMMON /TRIG/   PI,RAD,PIO2
506      COMMON /GEOID/  ERADPL, ERECSQ
507      COMMON /SNGLR/  SALT,DISOUT,GCLATD,GDLATD,GLOND,GAZD,GDZED,
508      *                  RY1,RY2,RY3,RHT,TSTEP
509      COMMON /SNGLI/   LIMIT,NTRAJC,IERRPT
510 C
511 C.....+.....+.....+.....+.....+...
512 C
513      DIMENSION P(6),Q(6),R(6),S(6),YB(6),FOLD(6),YOLD(6)
514 C
515 C.....+.....+.....+.....+.....+...
516 C
517      CHARACTER*1 CF,CR
518      CHARACTER*6 CNAME
519 C
520      DATA CF,CR / 'F','R'/
521      DATA CNAME / ' I95 '/                                ###
522 C
523 C.....+.....+.....+.....+.....+...
524 C
525      IF (IERRPT.GT.0) WRITE (16,2000) PC,INDXPC,RY1,RY2,RY3
526      2000 FORMAT (' SINGLTJ ',F8.3,I8,3F8.4)
527 C
528      BETAST = 2.0
529      LSTEP = 0
530      KBF = 0
531 C
532 C.....+.....+.....+.....+.....+...
533 C      \/ Runge-Kutta constants
534 C.....+.....+.....+.....+.....+...
535 C
536      RC106 = 1.0/6.0
537      SR2 = SQRT(2.0)
538      TMS202 = (2.0-SR2)/2.0
539      TPS202 = (2.0+SR2)/2.0
540 C
541 C.....+.....+.....+.....+.....+...
542 C      \/ Initialize Runge-Kutta variables to zero
543 C.....+.....+.....+.....+.....+...
544 C
545      100 DO 110 I = 1, 6
546          YB(I) = 0.0
547          S(I) = 0.0
548          R(I) = 0.0
549          Q(I) = 0.0
550          P(I) = 0.0
551          F(I) = 0.0
552      110 CONTINUE
553 C
554      NMAX = 0
555      NMIN = 0
556      NSTEP = 0
557      NSTEPT = 0
558 C
559      TAU = 0.0
560      TU100 = 0.0
```

```

561      YMAX = RY1
562      C
563      C.....+.....+.....+.....+.....+.....+...
564      C     \/ Define initial point at start of trajectory
565      C.....+.....+.....+.....+.....+.....+...
566      C
567      Y(1) = RY1
568      Y(2) = RY2
569      Y(3) = RY3
570      GRNDKM = (ERADPL/SQRT(1.0-ERECSQ*TSY2SQ))
571      Y10 = (RHT+GRNDKM)/ERAD
572      R120KM = (ERAD+120.0)/ERAD
573      C
574      C.....+.....+.....+.....+.....+.....+...
575      C     Rigidity = momentum/charge
576      C           use oxygen 16 as reference isotope
577      C           Constants used from Handbook of Physics (7-170)
578      C           1 amu = 0.931141 GeV
579      C.....+.....+.....+.....+.....+...
580      C
581      ANUC = 16.0
582      ZCHARGE = 8.0
583      C
584      EMCSQ = 0.931141
585      TENG = SQRT((PC*ZCHARGE)**2+(ANUC*EMCSQ)**2)
586      EOMC = -8987.566297*ZCHARGE/TENG
587      GMA = SQRT(((PC*ZCHARGE)/(EMCSQ*ANUC))**2+1.0)
588      BETA = SQRT(1.0-1.0/(GMA*GMA))
589      PVEL = VEL*BETA
590      HMAX = 1.0/PVEL
591      DISCK = DISCK - 1.1*hmax*pvel
592      C
593      C.....+.....+.....+.....+.....+.....+...
594      C     \/ Set max step length ("HMAX") to 1 earth radii
595      C           PVEL is particle velocity in earth radii per second
596      C           DISCK is check for approaching termination boundary
597      C           (within 1.1 steps)
598      C.....+.....+.....+.....+.....+...
599      C
600      EDIF = BETA*1.0E-4
601      if (edif.lt.1.0-5) edif = 1.0e-5
602      if (beta.lt.0.1) edif = 1.0e-4
603      C
604      Y(4) = BETA*Y1GC
605      Y(5) = BETA*Y2GC
606      Y(6) = BETA*Y3GC
607      C
608      azd = gdazd
609      zed = gdzed
610      IAZ = AZD+0.01
611      IZE = ZED+0.01
612      C
613      C.....+.....+.....+.....+.....+.....+...
614      C     \/ Set HSTART to about 1 % of the time to complete one gyro-radius
615      C           in a 1 Gauss field
616      C           H = [(2.0*PI*33.333*PC)/(BETA*C)]/0.01
617      C           if restart after BETA error, set HCK to small value
618      C           Introduce AHLT to control step size at high lat (beta problem)
619      C           HCK - reduce step size when large acceleration
620      C           HOLD - last step size used
621      C           HCNG - only allow 20% max growth in step size
622      C           HSNEK - attempt to approach boundary quickly
623      C           Problem at z=90 at high lat
624      C           add zen angle in deg to reduce first step
625      C.....+.....+.....+.....+.....+...
626      C
627      PTCY2 = ABS(TCY2)
628      AHLT = (1.0 + PTCY2)**2
629      HSTART = 6.0E-6*PC/(BETA*AHLT + ZED*PTCY2)
630      IF (HSTART.LT.1.0E-6) HSTART = 1.0E-6

```

```

631      HOLD = HSTART
632      HCK = HSTART
633      HCNG = HSTART
634      C
635      C      WRITE (16, 2010) HMAX,HOLD,HCK,HCNG,Y(4),Y(5),Y(6),PVEL, NSTEP
636      C2010 FORMAT (' 2010 ',18X, 4F9.6, 3F9.4, F9.4,9X,15X,I6)
637      C
638      C.....+.....+.....+.....+.....+.....+.....+...
639      C      Start Runge-Kutta
640      C      \\\//\//\//\//\//\//
641      C      \\\//\//\//\//
642      C      \/\//\//\/
643      C      \/\/
644      C      \
645      C.....+.....+.....+.....+.....+.....+...
646      C      Change in step size criteria, Aug 97
647      C      remove cos VxB step size, causes problems in tight loops
648      C      step size is now only a function of B and BETA
649      C.....+.....+.....+.....+.....+...
650      C
651      130 IF (HCK.LT.1.0E-6) HCK = 1.0E-6
652      CALL FGRAD
653      HB = 1.6E-5*PC/(B*BETA)
654      H = HB/BETAST
655      C
656      IF (KBF.GT.0) H=H/(FLOAT(KBF*2))
657      IF (H.GT.HMAX) H = HMAX
658      IF (H.GT.HCNG) H = HCNG
659      IF (H.GT.HCK) H = HCK
660      C
661      DO 140 I = 1, 6
662      S(I) = H*F(I)
663      P(I) = 0.5*S(I)-Q(I)
664      YB(I) = Y(I)
665      Y(I) = Y(I)+P(I)
666      R(I) = Y(I)-YB(I)
667      Q(I) = Q(I)+3.0*R(I)-0.5*S(I)
668      140 CONTINUE
669      C
670      CALL FGRAD
671      C
672      DO 150 I = 1, 6
673      S(I) = H*F(I)
674      P(I) = TMS2O2*(S(I)-Q(I))
675      YB(I) = Y(I)
676      Y(I) = Y(I)+P(I)
677      R(I) = Y(I)-YB(I)
678      Q(I) = Q(I)+3.0*R(I)-TMS2O2*S(I)
679      150 CONTINUE
680      C
681      CALL FGRAD
682      C
683      DO 160 I = 1, 6
684      S(I) = H*F(I)
685      P(I) = TPS2O2*(S(I)-Q(I))
686      YB(I) = Y(I)
687      Y(I) = Y(I)+P(I)
688      R(I) = Y(I)-YB(I)
689      Q(I) = Q(I)+3.0*R(I)-TPS2O2*S(I)
690      160 CONTINUE
691      C
692      CALL FGRAD
693      C
694      DO 170 I = 1, 6
695      S(I) = H*F(I)
696      P(I) = RC1O6*(S(I)-2.0*Q(I))
697      YB(I) = Y(I)
698      Y(I) = Y(I)+P(I)
699      R(I) = Y(I)-YB(I)
700      Q(I) = Q(I)+3.0*R(I)-0.5*S(I)

```

```
701      170 CONTINUE
702      C
703      C.....+.....+.....+.....+.....+.....+...
704      C      /\
705      C      /\ \
706      C      /\ /\ /\ \
707      C      /\ /\ /\ /\ /\ \
708      C      /\ /\ /\ /\ /\ /\ \
709      C      One Runge-Kutta
710      C      step completed
711      C.....+.....+.....+.....+.....+...
712      C
713      NSTEP = NSTEP+1
714      NSTEPT = NSTEPT + 1
715      TAU = TAU+H
716      HOLD = H
717      HCNG = H*1.2
718      HCK = HCNG
719      C
720      C.....+.....+.....+.....+.....+...
721      C      \/ Emergency diagnostic printout if desired
722      C.....+.....+.....+.....+.....+...
723      C      WRITE (16, 2030)      H,      Y(1),Y(2),Y(3),      PVEL,B, NSTEP
724      C      WRITE (16, 2040)      HB,H,HMAX,HOLD,HCK,HCNG,Y(4),Y(5),Y(6),
725      C      *                      PVEL,B,NSTEP
726      C2030 FORMAT (' 2030 ', 9X,      F9.6, 36X,      3F9.5,F9.4,F9.5,18X,I6)
727      C2040 FORMAT (' 2040 ',           6F9.6,            3F9.5,F9.4,F9.5,18X,I6)
728      C
729      C.....+.....+.....+.....+...
730      C      \/ Check for altitude less than 100 km
731      C      if less than 120 km, compute exact altitude above oblate earth
732      C      and sum time trajectory is below 100 km altitude.
733      C      set re-entrant altitude at RHT km above oblate earth
734      C      computed from international reference ellipsoid
735      C
736      C
737      IF (Y(1).LT.R120KM) THEN
738          TSY2SQ = SIN(Y(2))**2
739          GRNDKM = (ERADPL/SQRT(1.0-ERECSQ*TSY2SQ))
740          R100KM = (100.0+GRNDKM)/ERAD
741          R120KM = (120.0+GRNDKM)/ERAD
742          IF (Y(1).LT.R100KM) TU100 = TU100+H
743          PSALT = Y(1)*ERAD-GRNDKM
744          Y10 = (RHT+GRNDKM)/ERAD
745      C
746          IF (NSTEP.GT.5) THEN
747              IF (Y(1).LT.Y10.OR.PSALT.LE.0.0) THEN
748                  IF (IERRPT.GT.2) WRITE (16, 2045) PSALT, Y(1), Y10
749                  IRT = -1
750                  GO TO 260
751          ENDIF
752      ENDIF
753  ENDIF
754  2045 FORMAT (' 2045 PSALT,Y(1),Y10',F10.6,1PE14.6,E14.6)
755  C
756  C.....+.....+.....+.....+...
757  C      \/ Begin error checks
758  C      (1) Check for unacceptable changes in BETA
759  C
760  C
761  RCKBETA = SQRT(Y(4)*Y(4)+Y(5)*Y(5)+Y(6)*Y(6))
762  TBETA = BETA-RCKBETA
763  IF (ABS(TBETA).GT.EDIF) THEN
764      KBF = KBF+1
765      BETAST = BETAST + AHLT
766      EDIF = 2.0*EDIF
767      IF (RCKBETA.GT.(1.0+EDIF)) THEN
768          BETAST = BETAST+FLOAT(KBF)*(1.0+AHLT)
769          WRITE (*,2050) KBF,BETA,RCKBETA
770          WRITE (*,2060) Y,H,PC,NSTEP
771
```

```

771      WRITE (16,2050) KBF,BETA,RCKBETA
772      WRITE (16,2060) Y,H,PC,NSTEP
773      ENDIF
774  C
775      WRITE (16,2070) PC,BETA,KBF,RCKBETA,NSTEP,TBETA,Y,H
776      WRITE (*,2070) PC,BETA,KBF,RCKBETA,NSTEP,TBETA,Y,H
777  C
778  C      +.....+.....+.....+.....+.....+.....+...
779  C      \/ Check for irrecoverable beta error
780  C      if KBF > 4, set fate to failed and start next rigidity
781  C      +.....+.....+.....+.....+.....+...
782  C
783      IF (KBF.lt.5) THEN
784          GO TO 100
785      ELSE
786          IRT = 0
787          PATH = -PVEL*TAU
788          ISALT = SALT+0.0001
789          WRITE (*,2080)
790          GO TO 280
791      ENDIF
792      ENDIF
793  C
794      2050 FORMAT (' 2050 ','KBF=',i5,5x,' error, the velocity of the ',
795      *      'particle (BETA) has exceeded the velocity of light'
796      *      ' BETA at start of trajectory was ',F10.7/
797      *      ' BETA now equals           ',F10.7/
798      *      ' reduce step size and try again   ')
799      2060 FORMAT (' 2060 ','Y,H,PC,NSTEP=,8F12.6,I10)
800      2070 FORMAT (' 2070 ','ERROR, Particle BETA changed;',
801      *      ' PC = ',F10.6,' GV'
802      *      6x,' Beta at start was ',F10.7,17X,'KBF=',I6/
803      *      6x,' Beta new equals  ',F10.7,10X,'step number',I6/
804      *      6x,' Beta difference  ',F10.7/
805      *      6x,' step length reduced and trajectory recalculated '
806      *      6x,'Y=',6F12.6,6X,' H=',F15.7)
807      2080 FORMAT (' 2080 ','Irrecoverable BETA error problem'/6x,
808      *      ' Set fate to Failed & make path length negative'/6x,
809      *      ' Terminate this trajectory & continue with program')
810  C
811  C      ...+.....+.....+.....+.....+.....+...
812  C      \/ Continue status checks
813  C      (2) Compute composite acceleration
814  C
815  C
816      ACCER = SQRT(F(4)*F(4)+F(5)*F(5)+F(6)*F(6))
817  C
818      IF (IERRPT.GT.3) WRITE (16,2090) Y, F, ACCER, H, NSTEP
819      2090 FORMAT (' Y,F,A,H ',f7.4,5f7.3,1x,1pe8.1,5e8.1,1x,e8.1, e9.2,I6)
820  C
821  C      ...+.....+.....+.....+.....+.....+...
822  C      \/ Continue status checks, make adjustment latitude dependant
823  C      (3) Monitor change in composite acceleration
824  C          If composite acceleration (new-old) change > 5
825  C          If composite acceleration (new/old) ratio > 2
826  C          change step size to a smaller value
827  C
828  C
829      IF (NSTEP.GE.2) THEN
830          IF (ACCER.GT.ACCLD) THEN
831              DELACC = ACCER-ACCLD
832              IF (DELACC.GT.5.0) THEN
833                  HCK = HCK/(1.0+AHLT)
834                  IF (IERRPT.GT.2) WRITE (16,2100)
835                  *                      H,HCK,y(1),DELACC,PC,NSTEP
836
837          RFA = ACCER/ACCLD
838          IF (RFA.GT.2.0) THEN
839              HCK = HCK/(1.0+AHLT)
840              IF (IERRPT.GT.2) WRITE (16,2110)
841                  *                      H,HCK,Y(1),RFA,PC,NSTEP

```

```

841           ENDFIF
842       ENDIF
843   ENDIF
844 C
845   2100  FORMAT (' 2100 ', 'H-REDUCE', 2X, 'H=', F8.6, 2X, 'HCK=', F8.6, 2X,
846     *      'Y(1)=', F7.4, 2X, 'DELACC=', F6.2, 4X, 'PC=', F8.3, 4X, 'NSTEP=', I8)
847   2110  FORMAT (' 2110 ', 'H-REDUCE', 2X, 'H=', F8.6, 2X, 'HCK=', F8.6, 2X,
848     *      'Y(1)=', F7.4, 4X, 'RFA=', F6.2, 4X, 'PC=', F8.3, 4X, 'NSTEP=', I8)
849 C
850 C   ...+.....+.....+.....+.....+.....+...
851 C   \/ Continue status checks, make adjustment latitude dependant
852 C       (4) Monitor change in acceleration components
853 C           If change in any acceleration component is more than
854 C               a factor of 3, reduce step length
855 C   ...+.....+.....+.....+.....+...
856 C
857   DO 200 ICK = 4, 6
858     AFOLD = ABS(FOLD(ICK))
859     IF (AFOLD.GT.3.0) THEN
860       RFCK = ABS(F(ICK)/AFOLD)
861       IF (RFCK.GT.3.0) THEN
862         HCK = HCK/(1.0+AHLT)
863         IF (IERRPT.GT.2) THEN
864           WRITE (16,2120) H,HCK,Y(1),NMAX,ICK,F(ICK),
865             ICK,FOLD(ICK),PC,NSTEP
866     &
867       ENDIF
868     ENDIF
869   200  CONTINUE
870   ENDIF
871 C
872   2120 FORMAT (' 2120 ', 'H-reduce', 2X, 'H=', F8.6, 2X, 'HCK=', F8.6, 2X,
873     *      'Y(1)=', F7.4, 2X, 'NAMX=', I4, 2X, 'F(' , I1, ')=' , F6.2, 2X,
874     *      'FOLD(' , I1, ')=' , F6.2, 2X, 'PC=', F6.3, 2X, 'NSTEP=', I6)
875 C
876   ACCOLD = ACCER
877 C
878 C   ...+.....+.....+.....+.....+...
879 C   \/ Error checks complete
880 C
881 C   \/ Find if a max or a min has occurred
882 C   ...+.....+.....+.....+...
883 C
884   IF (NSTEP.GT.1) THEN
885     IF (YOLD(4).LE.0.0.AND.Y(4).GT.0.0) NMIN = NMIN+1
886     IF (YOLD(4).GE.0.0.AND.Y(4).LT.0.0) NMAX = NMAX+1
887   ENDIF
888 C
889   IF (Y(1).GT.YMAX) YMAX = Y(1)
890 C
891 C   ...+.....+.....+.....+...
892 C   \/ Check for termination conditions
893 C       Allowed - radial distance exceeded disout
894 C       Failed - number of steps exceeded
895 C       Re-entrant - trajectory is below "top" of atmosphere
896 C
897 C   ...+.....+.....+.....+...
898 C   \/ (1) Check for step limit exceeded
899 C   ...+.....+.....+...
900 C
901   IF (NSTEP.GE.LIMIT) THEN
902     IRT = 0
903     GO TO 260
904   ENDIF
905 C
906 C   ...+.....+.....+.....+...
907 C   \/ (2) Check if y(1) within 1.1 max step lengths of disout.
908 C           if so, reduce step size and
909 C               approach boundary at smaller step
910 C   ...+.....+.....+.....+...

```

```
911   C
912     IF (Y(1).GT.DISCK) THEN
913       DISTR = ABS(DISOUT - Y(1))
914       HSNEK = DISTR/PVEL
915       HCNG = HCNG/2.0
916       HCK = HCK/2.0
917       IF (HSNEK .LT. HCNG) HCNG = HSNEK
918       IF (HSNEK .LT. HCK) HCK = HSNEK
919       DISCK = DISOUT - DISTR/2.0
920       IF (DISCK.GE.DISOUT) THEN
921         DISCK = 24.999
922         GO TO 210
923     ENDIF
924     IF (H.LT.1.0E-5 .OR. HCK.LT.1.0E-5 .OR. HCNG.LT.1.0E-5) THEN
925       H = 1.0E-5
926       HCK = 1.0E-5
927       HCNG = 1.0E-5
928     ENDIF
929   C
930     IF (IERRPT.GT.3) WRITE (16,2130) Y(1),DISCK,PVEL,H,HSNEK,NSTEP
931   C
932     210  IF (Y(1).GT.DISOUT) THEN
933       IF (H.LE.1.0E-5) THEN
934         IRT = 1
935         GO TO 260
936       ENDIF
937       TAU = TAU - H
938       DO 220 I = 1, 6
939         Y(I) = YOLD(I)
940         F(I) = FOLD(I)
941     220  CONTINUE
942     endif
943     GO TO 130
944   ENDIF
945   2130 FORMAT (' 2130 ',2X,'Y(1),DISCK,PVEL,H,HSNEK',
946             *           4X,1PE12.6,4X,E12.6,4X,E12.6,4X,2E9.2,22X,I6)
947   C
948   C +.....+.....+.....+.....+.....+.....+...
949   C   \/\ Have penetrated boundary if you are here.
950   C   if large step size, go back one step and
951   C   reduce step length (and adjust "TAU")
952   C +.....+.....+.....+.....+.....+...
953   C
954   230  IF (Y(1).GT.DISOUT) THEN
955   C
956     IF (IERRPT.GT.3) WRITE (16, 2140) y(1),disck,pvel,H,nstep
957   C
958     if (h.lt.1.0e-5 .or. hck.lt.1.0e-5 .or. hcng.lt.1.0e-5) then
959       IRT = 1
960       go to 260
961     else
962       hck = hck/2.0
963       hcng = hcng/2.0
964       TAU = TAU - H
965       DO 240 I = 1, 6
966         Y(I) = YOLD(I)
967         F(I) = FOLD(I)
968     240  CONTINUE
969     GO TO 130
970   ENDIF
971   ENDIF
972   C
973   2140 FORMAT (' 2140 ',2X,'y(1),disck,pvel,H',
974             *           4X,1PE12.6,4X,E12.6,4X,E12.6,4X,E9.2,27X,I6)
975   C
976   C +.....+.....+.....+.....+.....+...
977   C   \/\ Store values of Y and F as FOLD & YOLD
978   C +.....+.....+.....+.....+.....+...
979   C
980   DO 250 I = 1, 6
```

```

981      YOLD(I) = Y(I)
982      FOLD(I) = F(I)
983 250 CONTINUE
984 C
985      GO TO 130
986 C
987 C.....+.....+.....+.....+.....+.....+.....+.....+...
988 C*****+*****+*****+*****+*****+*****+*****+*****+...
989 C      *****+*****+*****+*****+*****+*****+*****+...
990 C      *****+*****+*****+*****+*****+*****+*****+...
991 C      *****+*****+*****+*****+*****+*****+*****+...
992 C      TRAJECTORY COMPLETE IF YOU ARE HERE
993 C.....+.....+.....+.....+.....+.....+...
994 C
995 260 CONTINUE
996 C
997 IF (Y(1).GE.DISOUT) IRT = 1
998 PATH = PVEL*TAU
999 ISALT = SALT+0.0001
1000 LSTEP = BETAST - 1.9
1001 C
1002 C.....+.....+.....+.....+.....+.....+.....+...
1003 C      \/ Write out results
1004 C      IRT      +1      ALLOWED          (FATE = 0)
1005 C      IRT      0       FAILED           (FATE = 2)
1006 C      IRT     -1      RE-ENTRANT      (FATE = 1)
1007 C.....+.....+.....+.....+.....+...
1008 C
1009 IF (IRT.GT.0) THEN
1010   TCY2 = COS(Y(2))
1011   TSY2 = SIN(Y(2))
1012   YDA5 = Y(5)*TCY2+Y(4)*TSY2
1013   ATRG1 = Y(4)*TCY2-Y(5)*TSY2
1014   ATRG2 = SQRT(Y(6)*Y(6)+YDA5*YDA5)
1015   FASLAT = 0.0
1016   IF (ATRG1.NE.0.0.AND.ATRG2.NE.0.0) FASLAT =
1017   *                               ATAN2(ATRG1,ATRG2)*RAD
1018   *                               FASLON = Y(3)*RAD
1019   *                               IF (Y(6).NE.0.0.AND.YDA5.NE.0.0) FASLON = (Y(3)+ATAN2(Y(6),
1020   *                                         YDA5))*RAD
1021   *                               IF (FASLON.LT.0.0) FASLON = FASLON+360.0
1022   *                               IF (FASLON.GT.360.0) FASLON = FASLON-360.0
1023 C
1024   WRITE (8,2150) GDLATD,GCLATD,GLOND,IZE,IAZ,PC,FASLAT,FASLON,
1025   *                  PATH,NMAX,NSTEP,TU100,YMAX,LSTEP,SALT,CNAME
1026 C
1027   IFATE = 0
1028   WRITE (7,2160) GDLATD,GLOND,PC,ZED,AZD,ISALT,FASLAT,FASLON,
1029   *                  NSTEP,IFATE,CNAME
1030   ENDIF
1031 2150 FORMAT (2F7.2,F9.2,I5,I4,F10.3,2F8.2,F11.5,I4,I7,F9.5,F9.4,
1032   *                 I4,F11.1,1X,A6,13X)
1033 2160 FORMAT (F7.2,F8.2,F9.3,2F6.1,I7,F7.2,F8.2,I7,3X,I3,3X,A6)
1034 C
1035   IF (IRT.LT.0) THEN
1036     RENLAT = (PIO2-Y(2))*RAD
1037     RENLON = Y(3)*RAD
1038 C
1039   WRITE (8,2170) GDLATD,GCLATD,GLOND,IZE,IAZ,PC,CR,CR,PATH,NMAX
1040   *                  ,NSTEP,TU100,YMAX,LSTEP,SALT,CNAME,RENLAT,RENLON
1041 C
1042   IFATE = 1
1043   WRITE (7,2180) GDLATD,GLOND,PC,ZED,AZD,ISALT,NSTEP,IFATE,CNAME
1044   ENDIF
1045 2170 FORMAT (2F7.2,F9.2,I5,I4,F10.3,5X,A1,2X,5X,A1,2X,F11.5,I4,I7,
1046   *                 F9.5,F9.4,I4,F11.1,1X,A6,F6.1,F7.1)
1047 2180 FORMAT (F7.2,F8.2,F9.3,2F6.1,I7,4X,'R',7X,'R',I9,3X,I3,3X,A6)
1048 C
1049   280 IF (IRT.EQ.0) THEN
1050   C

```

```

1051      WRITE (8,2190) GDLATD,GCLATD,GLOND,IZE,IAZ,PC,CF,CF,PATH,
1052      *          NMAX,NSTEP,TU100,YMAX,LSTEP,SALT,CNAME
1053      C
1054      IFATE = 2
1055      IF (YMAX.LT.6.6) IFATE = 3
1056      WRITE (7,2200) GDLATD,GLOND,PC,ZED,AZD,ISALT,NSTEP,IFATE,CNAME
1057      ENDIF
1058 2190 FORMAT (2F7.2,F9.2,I5,I4,F10.3,5X,A1,2X,5X,A1,2X,F11.5,I4,I7,
1059      *          F9.5,F9.5,I4,F11.1,1X,A6,13X)
1060 2200 FORMAT (F7.2,F8.2,F9.3,2F6.1,I7,4X,'F',7X,'F',I9,3X,I3,3X,A6)
1061      C
1062      NTRAJC = NTRAJC+1
1063      TSTEP = TSTEP+FLOAT(NSTEP)
1064      C
1065      C ...+.....+.....+.....+.....+.....+.....+...
1066      C \/ Comment out to reduce IO
1067      C ...+.....+.....+.....+.....+.....+...
1068      C
1069      C      WRITE (*,2210) PC, ZED, AZD, NSTEP, IFATE
1070 2210 FORMAT (1H+, 22X, 3F7.2,7X,2I6)
1071      C
1072      IRSLT = IRT
1073      RETURN
1074      END
1075      SUBROUTINE FGRAD
1076      C
1077      C.....+.....+.....+.....+.....+.....+.....+...
1078      C Mod Feb 96 standard reference TJ1V (line check 17 Feb)
1079      C Mod 27 Jan 1999 Change MAGNEW to NEWMAG95      ###
1080      C.....+.....+.....+.....+.....+.....+...
1081      C Programmer - Don F. Smart; FORTRAN77
1082      C Note - The programming adheres to the conventional FORTRAN
1083      C default standard that variables beginning with
1084      C 'i','j','k','l','m', or 'n' are integer variables
1085      C Variables beginning with "c" are character variables
1086      C All other variables are real
1087      C.....+.....+.....+.....+.....+.....+...
1088      C Do not mix different type variables in same common block
1089      C Some computers do not allow this
1090      C.....+.....+.....+.....+.....+...
1091      C
1092      IMPLICIT INTEGER (I-N)
1093      IMPLICIT REAL * 8 (A-B)
1094      IMPLICIT REAL * 8 (D-H)
1095      IMPLICIT REAL * 8 (O-Z)
1096      C
1097      C.....+.....+.....+.....+.....+...
1098      C
1099      COMMON /WRKVLU/ F(6),Y(6),ERAD,EOMC,VEL,BR,BT,BP,B
1100      COMMON /WRKTSC/ TSY2,TCY2,TSY3,TCY3
1101      C
1102      C.....+.....+.....+.....+.....+...
1103      C
1104      F(1) = VEL*Y(4)
1105      F(2) = VEL*Y(5)/Y(1)
1106      TSY2 = SIN(Y(2))
1107      TCY2 = COS(Y(2))
1108      F(3) = VEL*Y(6)/(Y(1)*TSY2)
1109      SQY6 = Y(6)*Y(6)/Y(1)
1110      Y5OY1 = Y(5)/Y(1)
1111      TAY2 = TSY2/TCY2
1112      CALL MAGNEW95      ###
1113      F(4) = EOMC*(Y(5)*BP-Y(6)*BT)+VEL*(Y(5)*Y5OY1+SQY6)
1114      F(5) = EOMC*(Y(6)*BR-Y(4)*BP)+VEL*(SQY6/TAY2-Y5OY1*Y(4))
1115      F(6) = EOMC*(Y(4)*BT-Y(5)*BR)-VEL*((Y5OY1*Y(6))/TAY2+Y(4)*Y(6)/
1116      *          Y(1))
1117      RETURN
1118      C
1119      C.....+.....+.....+.....+.....+...
1120      C          Y(1) is R coordinate          Y(2) is THETA coordinate

```

```

1121 C      Y(3) is PHI coordinate      Y(4) is V(R)
1122 C      Y(5) is V(THETA)          Y(6) is V(PHI)
1123 C      F(1) is R dot            F(2) is THETA dot
1124 C      F(3) is PHI dot          F(4) is R dot dot
1125 C      F(5) is THETA dot dot    F(6) is PHI dot dot
1126 C      BR   is B(R)           BT   is B(THETA)
1127 C      BP   is B(PHI)          B    is magnitude of magnetic field
1128 C.....+.....+.....+.....+.....+.....+.....+...
1129 C
1130 END                                     ###
1131 SUBROUTINE MAGNEW95
1132 C.....+.....+.....+.....+.....+.....+...
1133 C Compute Magnetic field
1134 C Derived from NASA (NSSDC) routine NEWMAG version of December 1965
1135 C modified for 10 order field
1136 C Coefficients for IGRF 1995 loaded into this subroutine      ###
1137 C Coefficients obtained from program CNGMAGN
1138 C.....+.....+.....+.....+.....+.....+...
1139 CLast Mod 27 Jan 1999 IGRF 95 coefficients
1140 C Mod Feb 1996 standard reference TJ1V (line check 17 Feb)
1141 C Mod Nov 1980 for arguments in labeled common
1142 C.....+.....+.....+.....+.....+.....+...
1143 C.....+.....+.....+.....+.....+.....+...
1144 C.....+.....+.....+.....+.....+...
1145 C Programmer - Don F. Smart; FORTRAN77
1146 C Note - The programming adheres to the conventional FORTRAN
1147 C default standard that variables beginning with
1148 C 'i', 'j', 'k', 'l', 'm', or 'n' are integer variables
1149 C Variables beginning with "c" are character variables
1150 C All other variables are real
1151 C.....+.....+.....+.....+.....+.....+...
1152 C Do not mix different type variables in same common block
1153 C Some computers do not allow this
1154 C.....+.....+.....+.....+...
1155 C
1156 IMPLICIT INTEGER (I-N)
1157 IMPLICIT REAL * 8(A-B)
1158 IMPLICIT REAL * 8(D-H)
1159 IMPLICIT REAL * 8(O-Z)
1160 C
1161 C.....+.....+.....+.....+.....+...
1162 C
1163 COMMON /WRKVLU/ F(6),Y(6),ERAD,EOMC,VEL,BR,BT,BP,B
1164 COMMON /WRKTSC/ TSY2,TCY2,TSY3,TCY3
1165 C
1166 C.....+.....+.....+.....+.....+...
1167 C
1168 DIMENSION G(11,11),BM(11)
1169 C
1170 C.....+.....+.....+.....+.....+...
1171 C
1172 C \ Load in data constants if this is the first time called
1173 C otherwise, skip to evaluation of magnetic field
1174 C designed to drop high order terms if contribution
1175 C would be less than "BERR"
1176 C also designed so the maximum order of expansion
1177 C can be specified
1178 C.....+.....+.....+.....+...
1179 C
1180 IF (JDATA.EQ.77) GO TO 120
1181 C
1182 C.....+.....+.....+.....+.....+...
1183 C Gauss normalized Schmidt coefficients ordered for fast computation
1184 C
1185 Cards for FORTRAN
1186 C 1995.00 Coef in CNGMAG format
1187 C DATA(G(N, 1),N=1,11)/ 0.000000E+00
1188 C, 0.296820E+05, 0.329550E+04, -0.332250E+04, -0.411687E+04
1189 C, 0.165375E+04, -0.952875E+03, -0.209137E+04, -0.120656E+04
1190 C, -0.379844E+03,

```

```

1191      c  0.541277E+03/
1192      DATA(G(N, 2),N=1,11)/ -0.531800E+04
1193      c,  0.178900E+04, -0.532432E+04,  0.694430E+04, -0.432758E+04
1194      c, -0.357864E+04, -0.120980E+04,  0.237646E+04, -0.268125E+03
1195      c, -0.114663E+04,
1196      c  0.973144E+03/
1197      DATA(G(N, 3),N=1,11)/  0.408071E+04
1198      c,  0.368061E+03, -0.145925E+04, -0.241868E+04, -0.113872E+04
1199      c, -0.182140E+04, -0.971375E+03, -0.289608E+02,  0.560824E+02
1200      c, -0.108650E-03,
1201      c -0.421384E+03/
1202      DATA(G(N, 4),N=1,11)/  0.805270E+03
1203      c, -0.584820E+03,  0.320971E+03, -0.607948E+03,  0.880585E+03
1204      c,  0.574158E+03,  0.171361E+04, -0.593873E+03,  0.372776E+03
1205      c,  0.995794E+03,
1206      c  0.826402E+03/
1207      DATA(G(N, 5),N=1,11)/ -0.144990E+04
1208      c,  0.907844E+03, -0.204982E+03,  0.222593E+03, -0.857832E+02
1209      c,  0.370495E+03, -0.109137E+02, -0.493957E+02,  0.374307E+03
1210      c, -0.507382E+03,
1211      c  0.233742E+03/
1212      DATA(G(N, 6),N=1,11)/ -0.447330E+03
1213      c, -0.120658E+04,  0.715344E+03,  0.141986E+03, -0.694545E+02
1214      c,  0.182406E+02, -0.395558E+02, -0.493957E+02, -0.593223E+02
1215      c,  0.134764E+03,
1216      c -0.295662E+03/
1217      DATA(G(N, 7),N=1,11)/  0.302450E+03
1218      c, -0.115071E+04, -0.667509E+03,  0.311041E+03, -0.930726E+01
1219      c, -0.188074E+02,  0.631392E+02, -0.242182E+02, -0.343261E+02
1220      c,  0.347959E+02,
1221      c -0.123960E+03/
1222      DATA(G(N, 8),N=1,11)/  0.273116E+04
1223      c,  0.724020E+03, -0.614352E+02, -0.271676E+03, -0.987915E+02
1224      c,  0.557020E+02,  0.194178E+01,  0.129452E+01,  0.000000E+00
1225      c, -0.527347E+02,
1226      c -0.200432E+02/
1227      DATA(G(N, 9),N=1,11)/ -0.804375E+03
1228      c,  0.112165E+04, -0.289937E+03,  0.561461E+03, -0.177967E+03
1229      c, -0.686523E+02,  0.426161E+02,  0.626707E+01,  0.438695E+01
1230      c,  0.000000E+00,
1231      c -0.245478E+02/
1232      DATA(G(N,10),N=1,11)/  0.242067E+04
1233      c, -0.162975E+04, -0.912811E+03,  0.394630E+03,  0.235837E+03
1234      c, -0.156581E+03, -0.527347E+02,  0.206718E+02, -0.609049E+00
1235      c,  0.365430E+01,
1236      c -0.796435E+01/
1237      DATA(G(N,11),N=1,11)/ -0.486572E+03
1238      c, -0.210692E+03, -0.495841E+03, -0.701225E+03,  0.295662E+03
1239      c,  0.000000E+00,  0.400864E+02, -0.245478E+02,  0.265478E+01
1240      c,  0.356177E+01,
1241      c  0.000000E+00/
1242      DATA JMAG/ 0/,MGNMAX/ 11/,GSUM/ -0.885846E+05/
1243      DATA BM/ 0.100078E+06
1244      c,  0.100078E+06,  0.396633E+05,  0.253449E+05,  0.134069E+05
1245      c,  0.684114E+04,  0.360156E+04,  0.197007E+04,  0.913524E+03
1246      c,  0.489229E+03,
1247      c  0.152640E+03/
1248      C 1995.00   Coef in CNGMAG format                                IGRF95
1249      C ****
1250      C * The array G contains Gauss normalized Schmidt coefficients
1251      C * the array G contains both the G and H coefficients
1252      C * G(1,1) = 0.0
1253      C * Schmidt G(N,M) corresponds to -G(NN+1,MM+1) Gauss normalized coef
1254      C * Schmidt H(N,M) corresponds to -G( MI ,NN+1) Gauss normalized coef
1255      C * where MI = M
1256      C ****
1257      C ****
1258      C ****
1259      IF (GMSUM.EQ.0) GO TO 110
1260      P22 = 0.

```

```

1261      BERR = 0.0001
1262      AR = 0.
1263      DO 100 L = 1, MGNMAX
1264          DO 100 M = 1, MGNMAX
1265              AR = AR+1.
1266              P22 = P22+AR*G(M,L)
1267      100 CONTINUE
1268      GMSUM = (GMSUM-P22)/GMSUM
1269      C
1270      C.....+.....+.....+.....+.....+.....+...
1271      C**** \ Note following print and stop statements
1272      C.....+.....+.....+.....+.....+...
1273      C
1274      IF (ABS(GMSUM).GT.1.E-4) THEN
1275          WRITE (*, 2200) GMSUM
1276          WRITE (7, 2200) GMSUM
1277          WRITE (8, 2200) GMSUM
1278          STOP
1279      ENDIF
1280      2200 FORMAT (' DATA WRONG IN MAGNEW',E15.6)
1281      C
1282      110 CONTINUE
1283      C
1284      GMSUM = 0.
1285      JDATA = 77
1286      C
1287      120 CONTINUE
1288      P21 = TCY2
1289      P22 = TSY2
1290      AR = 1.0/Y(1)
1291      C
1292      C.....+.....+.....+.....+.....+...
1293      C      \/ N= 2
1294      C.....+.....+.....+.....+...
1295      C
1296      DP22 = P21
1297      TSY3 = SIN(Y(3))
1298      TCY3 = COS(Y(3))
1299      TSP2 = TSY3
1300      TCP2 = TCY3
1301      DP21 = -P22
1302      AOR = AR*AR*AR
1303      RC2 = G(2,2)*TCP2+G(1,2)*TSP2
1304      BR = -(AOR+AOR)*(G(2,1)*P21+RC2*P22)
1305      BT = AOR*(G(2,1)*DP21+RC2*DP22)
1306      BP = AOR*(G(1,2)*TCP2-G(2,2)*TSP2)*P22
1307      C
1308      C.....+.....+.....+.....+...
1309      C      \/ N = 3
1310      C.....+.....+.....+...
1311      C
1312      IF (MGNMAX.LT.3) GO TO 130
1313      AOR = AOR*AR
1314      ERR = BERR*SQRT((BP/P22)**2+BR**2+BT**2)
1315      IF ((BM(3)*AOR).LT.ERR) GO TO 130
1316      TSP3 = (TSP2+TSP2)*TCP2
1317      TCP3 = (TCP2+TSP2)*(TCP2-TSP2)
1318      P31 = P21*P21-0.33333333
1319      P32 = P21*P22
1320      P33 = P22*P22
1321      DP31 = -P32-P32
1322      DP32 = P21*P21-P33
1323      DP33 = -DP31
1324      RC2 = G(3,2)*TCP2+G(1,3)*TSP2
1325      RC3 = G(3,3)*TCP3+G(2,3)*TSP3
1326      BR = BR-3.0*AOR*(G(3,1)*P31+RC2*P32+RC3*P33)
1327      BT = BT+AOR*(G(3,1)*DP31+RC2*DP32+RC3*DP33)
1328      BP = BP-AOR*((G(3,2)*TSP2-G(1,3)*TCP2)*P32+
1329      *      2.0*(G(3,3)*TSP3-G(2,3)*TCP3)*P33)
1330      C

```

```

1331 C.....+.....+.....+.....+.....+.....+...
1332 C \/ N = 4
1333 C.....+.....+.....+.....+.....+...
1334 C
1335 IF (MGNMAX.LT.4) GO TO 130
1336 AOR = AOR*AR
1337 IF ((BM(4)*AOR).LT.ERR) GO TO 130
1338 TSP4 = TSP2*TCP3+TCP2*TSP3
1339 TCP4 = TCP2*TCP3-TSP2*TSP3
1340 P41 = P21*P31-0.26666666*P21
1341 DP41 = P21*DP31+DP21*P31-0.26666666*DP21
1342 P42 = P21*P32-0.20000000*P22
1343 DP42 = P21*DP32+DP21*P32-0.20000000*DP22
1344 P43 = P21*P33
1345 DP43 = P21*DP33+DP21*P33
1346 P44 = P22*P33
1347 DP44 = 3.0*P43
1348 RC2 = G(4,2)*TCP2+G(1,4)*TSP2
1349 RC3 = G(4,3)*TCP3+G(2,4)*TSP3
1350 RC4 = G(4,4)*TCP4+G(3,4)*TSP4
1351 BR = BR-4.0*AOR*(G(4,1)*P41+RC2*P42+RC3*P43+RC4*P44)
1352 BT = BT+AOR*(G(4,1)*DP41+RC2*DP42+RC3*DP43+RC4*DP44)
1353 BP = BP-AOR*((G(4,2)*TSP2-G(1,4)*TCP2)*P42+
1354 * 2.0*(G(4,3)*TSP3-G(2,4)*TCP3)*P43+
1355 * 3.0*(G(4,4)*TSP4-G(3,4)*TCP4)*P44)
1356 C
1357 C.....+.....+.....+.....+.....+...
1358 C \/ N = 5
1359 C.....+.....+.....+.....+.....+...
1360 C
1361 IF (MGNMAX.LT.5) GO TO 130
1362 AOR = AOR*AR
1363 IF ((BM(5)*AOR).LT.ERR) GO TO 130
1364 TSP5 = (TSP3+TSP3)*TCP3
1365 TCP5 = (TCP3+TSP3)*(TCP3-TSP3)
1366 P51 = P21*P41-0.25714285*P31
1367 DP51 = P21*DP41+DP21*P41-0.25714285*DP31
1368 P52 = P21*P42-0.22857142*P32
1369 DP52 = P21*DP42+DP21*P42-0.22857142*DP32
1370 P53 = P21*P43-0.14285714*P33
1371 DP53 = P21*DP43+DP21*P43-0.14285714*DP33
1372 P54 = P21*P44
1373 DP54 = P21*DP44+DP21*P44
1374 P55 = P22*P44
1375 DP55 = 4.0*P54
1376 RC2 = G(5,2)*TCP2+G(1,5)*TSP2
1377 RC3 = G(5,3)*TCP3+G(2,5)*TSP3
1378 RC4 = G(5,4)*TCP4+G(3,5)*TSP4
1379 RC5 = G(5,5)*TCP5+G(4,5)*TSP5
1380 BR = BR-5.0*AOR*(G(5,1)*P51+RC2*P52+RC3*P53+RC4*P54+RC5*P55)
1381 BT = BT+AOR*(G(5,1)*DP51+RC2*DP52+RC3*DP53+RC4*DP54+RC5*DP55)
1382 BP = BP-AOR*((G(5,2)*TSP2-G(1,5)*TCP2)*P52+2.0*(G(5,3)*TSP3-
1383 * G(2,5)*TCP3)*P53+3.0*(G(5,4)*TSP4-
1384 * G(3,5)*TCP4)*P54+4.0*(G(5,5)*TSP5-
1385 * G(4,5)*TCP5)*P55)
1386 C
1387 C.....+.....+.....+.....+...
1388 C \/ N = 6
1389 C.....+.....+.....+.....+...
1390 C
1391 IF (MGNMAX.LT.6) GO TO 130
1392 AOR = AOR*AR
1393 IF ((BM(6)*AOR).LT.ERR) GO TO 130
1394 TSP6 = TSP2*TCP5+TCP2*TSP5
1395 TCP6 = TCP2*TCP5-TSP2*TSP5
1396 P61 = P21*P51-0.25396825*P41
1397 DP61 = P21*DP51+DP21*P51-0.25396825*DP41
1398 P62 = P21*P52-0.23809523*P42
1399 DP62 = P21*DP52+DP21*P52-0.23809523*DP42
1400 P63 = P21*P53-0.19047619*P43

```

```

1401      DP63 = P21*DP53+DP21*P53-0.19047619*DP43
1402      P64 = P21*P54-0.11111111*P44
1403      DP64 = P21*DP54+DP21*P54-0.11111111*DP44
1404      P65 = P21*P55
1405      DP65 = P21*DP55+DP21*P55
1406      P66 = P22*P55
1407      DP66 = 5.0*P65
1408      RC2 = G(6,2)*TCP2+G(1,6)*TSP2
1409      RC3 = G(6,3)*TCP3+G(2,6)*TSP3
1410      RC4 = G(6,4)*TCP4+G(3,6)*TSP4
1411      RC5 = G(6,5)*TCP5+G(4,6)*TSP5
1412      RC6 = G(6,6)*TCP6+G(5,6)*TSP6
1413      BR = BR-6.0*AOR*(G(6,1)*P61+RC2*P62+RC3*P63+RC4*P64+RC5*P65
1414      * +RC6*P66)
1415      BT = BT+AOR*(G(6,1)*DP61+RC2*DP62+RC3*DP63+RC4*DP64+RC5*DP65
1416      * +RC6*DP66)
1417      BP = BP-AOR*((G(6,2)*TSP2-G(1,6)*TCP2)*P62+2.0*(G(6,3)*TSP3
1418      * -G(2,6)*TCP3)*P63+3.0*(G(6,4)*TSP4
1419      * -G(3,6)*TCP4)*P64+4.0*(G(6,5)*TSP5
1420      * -G(4,6)*TCP5)*P65+5.0*(G(6,6)*TSP6-G(5,6)*TCP6)*P66)
1421      C
1422      C.....+.....+.....+.....+.....+.....+.....+...
1423      C \/ N = 7
1424      C.....+.....+.....+.....+.....+.....+...
1425      C
1426      IF (MGNMAX.LT.7) GO TO 130
1427      AOR = AOR*AR
1428      IF ((BM(7)*AOR).LT.ERR) GO TO 130
1429      TSP7 = (TSP4+TSP4)*TCP4
1430      TCP7 = (TCP4+TSP4)*(TCP4-TSP4)
1431      P71 = P21*P61-0.25252525*P51
1432      DP71 = P21*DP61+DP21*P61-0.25252525*DP51
1433      P72 = P21*P62-0.24242424*P52
1434      DP72 = P21*DP62+DP21*P62-0.24242424*DP52
1435      P73 = P21*P63-0.21212121*P53
1436      DP73 = P21*DP63+DP21*P63-0.21212121*DP53
1437      P74 = P21*P64-0.16161616*P54
1438      DP74 = P21*DP64+DP21*P64-0.16161616*DP54
1439      P75 = P21*P65-0.09090909*P55
1440      DP75 = P21*DP65+DP21*P65-0.09090909*DP55
1441      P76 = P21*P66
1442      DP76 = P21*DP66+DP21*P66
1443      P77 = P22*P66
1444      DP77 = 6.0*P76
1445      RC2 = G(7,2)*TCP2+G(1,7)*TSP2
1446      RC3 = G(7,3)*TCP3+G(2,7)*TSP3
1447      RC4 = G(7,4)*TCP4+G(3,7)*TSP4
1448      RC5 = G(7,5)*TCP5+G(4,7)*TSP5
1449      RC6 = G(7,6)*TCP6+G(5,7)*TSP6
1450      RC7 = G(7,7)*TCP7+G(6,7)*TSP7
1451      BR = BR-7.0*AOR*(G(7,1)*P71+RC2*P72+RC3*P73+RC4*P74+RC5*P75
1452      * +RC6*P76+RC7*P77)
1453      BT = BT+AOR*(G(7,1)*DP71+RC2*DP72+RC3*DP73+RC4*DP74+RC5*DP75
1454      * +RC6*DP76+RC7*DP77)
1455      BP = BP-AOR*((G(7,2)*TSP2-G(1,7)*TCP2)*P72+2.0*(G(7,3)*TSP3
1456      * -G(2,7)*TCP3)*P73+3.0*(G(7,4)*TSP4
1457      * -G(3,7)*TCP4)*P74+4.0*(G(7,5)*TSP5
1458      * -G(4,7)*TCP5)*P75+5.0*(G(7,6)*TSP6
1459      * -G(5,7)*TCP6)*P76+6.0*(G(7,7)*TSP7-G(6,7)*TCP7)*P77)
1460      C
1461      C.....+.....+.....+.....+.....+.....+...
1462      C \/ N = 8
1463      C.....+.....+.....+.....+.....+...
1464      C
1465      IF (MGNMAX.LT.8) GO TO 130
1466      AOR = AOR*AR
1467      IF ((BM(8)*AOR).LT.ERR) GO TO 130
1468      TSP8 = TSP2*TCP7+TCP2*TSP7
1469      TCP8 = TCP2*TCP7-TSP2*TSP7
1470      P81 = P21*P71-0.25174825*P61

```

```

1471      DP81 = P21*DP71+DP21*P71-0.25174825*DP61
1472      P82 = P21*P72-0.24475524*P62
1473      DP82 = P21*DP72+DP21*P72-0.24475524*DP62
1474      P83 = P21*P73-0.22377622*P63
1475      DP83 = P21*DP73+DP21*P73-0.22377622*DP63
1476      P84 = P21*P74-0.18881118*P64
1477      DP84 = P21*DP74+DP21*P74-0.18881118*DP64
1478      P85 = P21*P75-0.13986013*P65
1479      DP85 = P21*DP75+DP21*P75-0.13986013*DP65
1480      P86 = P21*P76-0.07692307*P66
1481      DP86 = P21*DP76+DP21*P76-0.07692307*DP66
1482      P87 = P21*P77
1483      DP87 = P21*DP77+DP21*P77
1484      P88 = P22*P77
1485      DP88 = 7.0*P87
1486      RC2 = G(8,2)*TCP2+G(1,8)*TSP2
1487      RC3 = G(8,3)*TCP3+G(2,8)*TSP3
1488      RC4 = G(8,4)*TCP4+G(3,8)*TSP4
1489      RC5 = G(8,5)*TCP5+G(4,8)*TSP5
1490      RC6 = G(8,6)*TCP6+G(5,8)*TSP6
1491      RC7 = G(8,7)*TCP7+G(6,8)*TSP7
1492      RC8 = G(8,8)*TCP8+G(7,8)*TSP8
1493      BR = BR-8.0*AOR*(G(8,1)*P81+RC2*P82+RC3*P83+RC4*P84+RC5*P85
1494      * +RC6*P86+RC7*P87+RC8*P88)
1495      BT = BT+AOR*(G(8,1)*DP81+RC2*DP82+RC3*DP83+RC4*DP84+RC5*DP85
1496      * +RC6*DP86+RC7*DP87+RC8*DP88)
1497      BP = BP-AOR*((G(8,2)*TSP2-G(1,8)*TCP2)*P82
1498      * +2.0*(G(8,3)*TSP3-G(2,8)*TCP3)*P83
1499      * +3.0*(G(8,4)*TSP4-G(3,8)*TCP4)*P84
1500      * +4.0*(G(8,5)*TSP5-G(4,8)*TCP5)*P85
1501      * +5.0*(G(8,6)*TSP6-G(5,8)*TCP6)*P86
1502      * +6.0*(G(8,7)*TSP7-G(6,8)*TCP7)*P87
1503      * +7.0*(G(8,8)*TSP8-G(7,8)*TCP8)*P88)
1504      C
1505      C.....+.....+.....+.....+.....+.....+...
1506      C \/ N = 9
1507      C.....+.....+.....+.....+.....+...
1508      C
1509      IF (MGNMAX.LT.9) GO TO 130
1510      AOR = AOR*AR
1511      IF ((BM(9)*AOR).LT.ERR) GO TO 130
1512      TSP9 = (TSP5+TSP5)*TCP5
1513      TCP9 = (TCP5+TSP5)*(TCP5-TSP5)
1514      P91 = P21*P81-0.25128205*P71
1515      DP91 = P21*DP81+DP21*P81-0.25128205*DP71
1516      P92 = P21*P82-0.24615384*P72
1517      DP92 = P21*DP82+DP21*P82-0.24615384*DP72
1518      P93 = P21*P83-0.23076923*P73
1519      DP93 = P21*DP83+DP21*P83-0.23076923*DP73
1520      P94 = P21*P84-0.20512820*P74
1521      DP94 = P21*DP84+DP21*P84-0.20512820*DP74
1522      P95 = P21*P85-0.16923076*P75
1523      DP95 = P21*DP85+DP21*P85-0.16923076*DP75
1524      P96 = P21*P86-0.12307692*P76
1525      DP96 = P21*DP86+DP21*P86-0.12307692*DP76
1526      P97 = P21*P87-0.06666666*P77
1527      DP97 = P21*DP87+DP21*P87-0.06666666*DP77
1528      P98 = P21*P88
1529      DP98 = P21*DP88+DP21*P88
1530      P99 = P22*P88
1531      DP99 = 8.0*P98
1532      RC2 = G(9,2)*TCP2+G(1,9)*TSP2
1533      RC3 = G(9,3)*TCP3+G(2,9)*TSP3
1534      RC4 = G(9,4)*TCP4+G(3,9)*TSP4
1535      RC5 = G(9,5)*TCP5+G(4,9)*TSP5
1536      RC6 = G(9,6)*TCP6+G(5,9)*TSP6
1537      RC7 = G(9,7)*TCP7+G(6,9)*TSP7
1538      RC8 = G(9,8)*TCP8+G(7,9)*TSP8
1539      RC9 = G(9,9)*TCP9+G(8,9)*TSP9
1540      BR = BR-9.0*AOR*(G(9,1)*P91+RC2*P92+RC3*P93+RC4*P94+RC5*P95

```

```

1541      *      +RC6*P96+RC7*P97+RC8*P98+RC9*P99)
1542      BT = BT+AOR*(G(9,1)*DP91+RC2*DP92+RC3*DP93+RC4*DP94+RC5*DP95
1543      *      +RC6*DP96+RC7*DP97+RC8*DP98+RC9*DP99)
1544      BP = BP-AOR*((G(9,2)*TSP2-G(1,9)*TCP2)*P92+2.0*(G(9,3)*TSP3
1545      *      -G(2,9)*TCP3)*P93+3.0*(G(9,4)*TSP4
1546      *      -G(3,9)*TCP4)*P94+4.0*(G(9,5)*TSP5
1547      *      -G(4,9)*TCP5)*P95+5.0*(G(9,6)*TSP6
1548      *      -G(5,9)*TCP6)*P96+6.0*(G(9,7)*TSP7
1549      *      -G(6,9)*TCP7)*P97+7.0*(G(9,8)*TSP8
1550      *      -G(7,9)*TCP8)*P98+8.0*(G(9,9)*TSP9-G(8,9)*TCP9)*P99)
1551      C
1552      C.....+.....+.....+.....+.....+.....+...
1553      C      \/ N = 10
1554      C.....+.....+.....+.....+.....+.....+...
1555      C
1556      IF (MGNMAX.LT.10) GO TO 130
1557      AOR = AOR*AR
1558      IF ((BM(10)*AOR).LT.ERR) GO TO 130
1559      TSP10 = TSP2*TCP9+TCP2*TSP9
1560      TCP10 = TCP2*TCP9-TSP2*TSP9
1561      P101 = P21*P91-0.25098039*P81
1562      DP101 = P21*DP91+DP21*P91-0.25098039*DP81
1563      P102 = P21*P92-0.24705882*P82
1564      DP102 = P21*DP92+DP21*P92-0.24705882*DP82
1565      P103 = P21*P93-0.23529411*P83
1566      DP103 = P21*DP93+DP21*P93-0.23529411*DP83
1567      P104 = P21*P94-0.21568627*P84
1568      DP104 = P21*DP94+DP21*P94-0.21568627*DP84
1569      P105 = P21*P95-0.18823529*P85
1570      DP105 = P21*DP95+DP21*P95-0.18823529*DP85
1571      P106 = P21*P96-0.15294117*P86
1572      DP106 = P21*DP96+DP21*P96-0.15294117*DP86
1573      P107 = P21*P97-0.10980392*P87
1574      DP107 = P21*DP97+DP21*P97-0.10980392*DP87
1575      P108 = P21*P98-0.05882352*P88
1576      DP108 = P21*DP98+DP21*P98-0.05882352*DP88
1577      P109 = P21*P99
1578      DP109 = P21*DP99+DP21*P99
1579      P1010 = P22*P99
1580      DP1010 = 9.0*P109
1581      RC2 = G(10,2)*TCP2+G(1,10)*TSP2
1582      RC3 = G(10,3)*TCP3+G(2,10)*TSP3
1583      RC4 = G(10,4)*TCP4+G(3,10)*TSP4
1584      RC5 = G(10,5)*TCP5+G(4,10)*TSP5
1585      RC6 = G(10,6)*TCP6+G(5,10)*TSP6
1586      RC7 = G(10,7)*TCP7+G(6,10)*TSP7
1587      RC8 = G(10,8)*TCP8+G(7,10)*TSP8
1588      RC9 = G(10,9)*TCP9+G(8,10)*TSP9
1589      RC10 = G(10,10)*TCP10+G(9,10)*TSP10
1590      BR = BR-10.0*AOR*(G(10,1)*P101+RC2*P102+RC3*P103+RC4*P104
1591      *      +RC5*P105+RC6*P106+RC7*P107+RC8*P108+RC9*P109+RC10*P1010)
1592      *      +RC5*P105+RC6*P106+RC7*P107+RC8*P108+RC9*P109
1593      *      +RC10*P1010)
1594      *      +RC10*P1010)
1595      BP = BP-AOR*((G(10,2)*TSP2-G(1,10)*TCP2)*P102+2.0*(G(10,3)*TSP3
1596      *      -G(2,10)*TCP3)*P103+3.0*(G(10,4)*TSP4
1597      *      -G(3,10)*TCP4)*P104+4.0*(G(10,5)*TSP5
1598      *      -G(4,10)*TCP5)*P105+5.0*(G(10,6)*TSP6
1599      *      -G(5,10)*TCP6)*P106+6.0*(G(10,7)*TSP7
1600      *      -G(6,10)*TCP7)*P107+7.0*(G(10,8)*TSP8
1601      *      -G(7,10)*TCP8)*P108+8.0*(G(10,9)*TSP9
1602      *      -G(8,10)*TCP9)*P109+9.0*(G(10,10)*TSP10
1603      *      -G(9,10)*TCP10)*P1010)
1604      C
1605      C.....+.....+.....+.....+.....+.....+...
1606      C      \/ N = 11
1607      C.....+.....+.....+.....+.....+.....+...
1608      C
1609      IF (MGNMAX.LT.11) GO TO 130
1610      AOR = AOR*AR

```

```

1611 IF ((BM(11)*AOR).LT.ERR) GO TO 130
1612 TSP11 = (TSP6+TSP6)*TCP6
1613 TCP11 = (TCP6+TSP6)*(TCP6-TSP6)
1614 P111 = P21*P101-0.25077399*P91
1615 DP111 = P21*DP101+DP21*P101-0.25077399*DP91
1616 P112 = P21*P102-0.24767801*P92
1617 DP112 = P21*DP102+DP21*P102-0.24767801*DP92
1618 P113 = P21*P103-0.23839009*P93
1619 DP113 = P21*DP103+DP21*P103-0.23839009*DP93
1620 P114 = P21*P104-0.22291021*P94
1621 DP114 = P21*DP104+DP21*P104-0.22291021*DP94
1622 P115 = P21*P105-0.20123839*P95
1623 DP115 = P21*DP105+DP21*P105-0.20123839*DP95
1624 P116 = P21*P106-0.17337461*P96
1625 DP116 = P21*DP106+DP21*P106-0.17337461*DP96
1626 P117 = P21*P107-0.13931888*P97
1627 DP117 = P21*DP107+DP21*P107-0.13931888*DP97
1628 P118 = P21*P108-0.09907120*P98
1629 DP118 = P21*DP108+DP21*P108-0.09907120*DP98
1630 P119 = P21*P109-0.05263157*P99
1631 DP119 = P21*DP109+DP21*P109-0.05263157*DP99
1632 P1110 = P21*P1010
1633 DP1110 = P21*DP1010+DP21*P1010
1634 P1111 = P22*P1010
1635 DP1111 = 10.0*P1110
1636 RC2 = G(11,2)*TCP2+G(1,11)*TSP2
1637 RC3 = G(11,3)*TCP3+G(2,11)*TSP3
1638 RC4 = G(11,4)*TCP4+G(3,11)*TSP4
1639 RC5 = G(11,5)*TCP5+G(4,11)*TSP5
1640 RC6 = G(11,6)*TCP6+G(5,11)*TSP6
1641 RC7 = G(11,7)*TCP7+G(6,11)*TSP7
1642 RC8 = G(11,8)*TCP8+G(7,11)*TSP8
1643 RC9 = G(11,9)*TCP9+G(8,11)*TSP9
1644 RC10 = G(11,10)*TCP10+G(9,11)*TSP10
1645 RC11 = G(11,11)*TCP11+G(10,11)*TSP11
1646 BR = BR-11.0*AOR*(G(11,1)*P111+RC2*P112+RC3*P113+RC4*P114
1647 * +RC5*P115+RC6*P116+RC7*P117+RC8*P118+RC9*P119+RC10*P1110
1648 * +RC11*P1111)
1649 BT = BT+AOR*(G(11,1)*DP111+RC2*DP112+RC3*DP113+RC4*DP114
1650 * +RC5*DP115+RC6*DP116+RC7*DP117+RC8*DP118+RC9*DP119
1651 * +RC10*DP1110+RC11*DP1111)
1652 BP = BP-AOR*((G(11,2)*TSP2-G(1,11)*TCP2)*P112+2.0*(G(11,3)*TSP3
1653 * -G(2,11)*TCP3)*P113 + 3.0 *(G(11,4)*TSP4
1654 * -G(3,11)*TCP4)*P114 + 4.0 *(G(11,5)*TSP5
1655 * -G(4,11)*TCP5)*P115 + 5.0 *(G(11,6)*TSP6
1656 * -G(5,11)*TCP6)*P116 + 6.0 *(G(11,7)*TSP7
1657 * -G(6,11)*TCP7)*P117 + 7.0 *(G(11,8)*TSP8
1658 * -G(7,11)*TCP8)*P118 + 8.0 *(G(11,9)*TSP9
1659 * -G(8,11)*TCP9)*P119 + 9.0 *(G(11,10)*TSP10
1660 * -G(9,11)*TCP10)*P1110+10.0*(G(11,11)*TSP11
1661 * -G(10,11)*TCP11)*P1111)

1662 C
1663 C.....+.....+.....+.....+.....+.....+.....+...
1664 C \/ Convert to units of Gauss
1665 C.....+.....+.....+.....+.....+.....+...
1666 C
1667 130 BP = BP/P22*1.E-5
1668 BT = BT*1.E-5
1669 BR = BR*1.E-5
1670 B = SQRT(BR*BR+BT*BT+BP*BP)
1671 RETURN
1672 C
1673 C.....+.....+.....+.....+.....+.....+...
1674 C Y(1) is R coordinate Y(2) is THETA coordinate
1675 C Y(3) is PHI coordinate Y(4) is V(R)
1676 C Y(5) is V(THETA) Y(6) is V(PHI)
1677 C F(1) is R dot F(2) is THETA dot
1678 C F(3) is PHI dot F(4) is R dot dot
1679 C F(5) is THETA dot dot F(6) is PHI dot dot
1680 C BR is B(R) BT is B(THETA)

```

1681 C BP is B(PHI) B is magnitude of magnetic field
1682 C.....+.....+.....+.....+.....+.....+...
1683 C
1684 END

```

1      PROGRAM TJALLMAG
2      C
3      C.....+.....+.....+.....+.....+.....+.....+...
4      c     Mod 08 Dec 00  Use ALLMAG for magnetic field calculations
5      C.....+.....+.....+.....+.....+.....+.....+...
6      C     Multi-platform COSMIC-RAY TRAJECTORY PROGRAM
7      C     FORTRAN 77 transportable version (no time functions)
8      C     Read in control card; LAT, LON, RIG, ZENITH, AZIMUTH, DELPC, INDO
9      C     Then calculate      INDO      trajectories
10     C           Starting at    PC
11     C           Incrementing at DELPC intervals
12     C     Includes conversion from Geodetic to Geocentric coordinates
13     C     Includes re-entrant albedo calculations
14     C     Uses subroutine SINGLE to do trajectory calculations
15     C     Magnetic field - as determined by the NASA ALLMAG routine
16     C.....+.....+.....+.....+.....+.....+...
17     C     Restrictions: Cannot run over N or S pole; will get BETA blowup
18     C.....+.....+.....+.....+.....+...
19     C     Mod History
20     Clast Mod 22 Dec 00  Use NASA ALLMAG for magnetic field calculations
21     C     Mod 21 Dec 00  Make all intrinsic function double precision for PC
22     C     Mod 20 Dec 00  Insert 8 character format 1000 with AZ & ZE
23     C     Mod 17 Feb 99  if (ymax.lt.6.6) IFATE = 3
24     C     Mod 17 Feb 99  set limit to 600000
25     C     Mod Aug 97   Adjust step size to minimize beta problems
26     C     Mod Jan 97   High latitude step size adjust, introduce AHLT
27     C     Mod Jun 96   EDIF limit set to 1.0e-5
28     C     Mod Jun 96   IERRPT formats, Boundary and look ahead
29     C     Mod May 96   Make all FORTRAN coding upper case
30     C     Mod Feb 96   Standard reference TJ1V line check
31     C
32     C     ***** Timing estimates base on COMPAQ Digital FORTRAN
33     C     Will run on PIII PC at 850 MHZ      55000 steps/sec (Real*8)
34     C     Will run on PIII PC at 700 MHZ      39000 steps/sec (Real*8)
35     C     Will run on PIII PC at 550 MHZ      32000 steps/sec (Real*8)
36     C     Will run on PII PC at 400 MHZ      23000 steps/sec (Real*8)
37     C
38     C     * TAPE*      Monitor program operation
39     C     * TAPE1      Trajectory control cards
40     C     * TAPE7      80 character card image output
41     C     * TAPE8      132 character line printer output
42     C     * TAPE16     Diagnostic output for trouble shooting
43     C     *          Normally turned off (open statement commented out)
44     C
45     C
46     C.....+.....+.....+.....+.....+.....+...
47     C     Programmer - Don F. Smart; FORTRAN77
48     C     Note - The programming adheres to the conventional FORTRAN
49     C           default standard that variables beginning with
50     C           'i','j','k','l','m',or 'n' are integer variables
51     C           Variables beginning with "c" are character variables
52     C           All other variables are real
53     C.....+.....+.....+.....+.....+...
54     C           Do not mix different type variables in same common block
55     C           Some computers do not allow this
56     C.....+.....+.....+.....+.....+...
57     C
58     IMPLICIT INTEGER (I-N)
59     IMPLICIT REAL * 8(A-B)
60     IMPLICIT REAL * 8(D-H)
61     IMPLICIT REAL * 8(O-Z)
62     C
63     C.....+.....+.....+.....+.....+...
64     C
65     COMMON /WRKVLU/ F(6),Y(6),ERAD,EOMC,VEL,BR,BT,BP,B
66     COMMON /WRKTSC/ TSY2,TCY2,TSY3,TCY3
67     COMMON /TRIG/ PI,RAD,PIO2
68     COMMON /GEOID/ ERADPL, ERECSQ
69     COMMON /SNGLR/ SALT,DISOUT,GCLATD,GDLATD,GLOND,GDAZD,GDZED,
70     *               RY1,RY2,RY3,RHT,TSTEP

```

```

71      COMMON /SNGLI/  LIMIT,NTRAJC,IERRPT
72      C
73      C.....+.....+.....+.....+.....+.....+...
74      C
75      OPEN (1, FILE='TAPE1', STATUS='OLD')
76      OPEN (7, FILE='TAPE7', STATUS='UNKNOWN')
77      OPEN (8, FILE='TAPE8', STATUS='UNKNOWN')
78      OPEN (16,FILE='TAPE16',STATUS='UNKNOWN')
79      C
80      C.....+.....+.....+.....+.....+.....+...
81      C  \/ User defined program control
82      C.....+.....+.....+.....+.....+.....+...
83      C
84      FSTEP = 4.0E08
85      LIMIT  = 600000
86      C
87      C.....+.....+.....+.....+.....+.....+...
88      C  \/ FSTEP  is total number of steps before run is terminated
89      C      LIMIT  is max number of steps before trajectory declared F
90      C.....+.....+.....+.....+.....+.....+...
91      C  \/ Define program constants
92      C.....+.....+.....+.....+.....+.....+...
93      C      DISOUT is radial distance for trajectory termination
94      C      ERAD   is average earth radius
95      C      NTRAJC is number of trajectory computed in this run
96      C      RHT    is top of atmosphere for re-entrant trajectory
97      C      TSTEP  is number of steps executed in this run
98      C.....+.....+.....+.....+.....+.....+...
99      C
100     NTRAJC = 0
101     TSTEP  = 0.0
102     C
103     DISOUT = 25.0
104     ERAD   = 6371.2
105     RHT    = 20.0
106     VEL    = 2.99792458E5/ERAD
107     C
108     C.....+.....+.....+.....+.....+.....+...
109     C      "VEL" is light velocity in earth radii per second
110     C      Light speed defined as 299792458 m/s
111     C      Ref: E. R. Cohen AND B. N. Taylor, "The Fundamental Physical
112     C      Constants, Physics Today P11, August 1987.
113     C.....+.....+.....+.....+.....+...
114     C  \/ Define essential trigonometric values
115     C.....+.....+.....+.....+.....+...
116     C
117     C      PI    = ACOS(-1.0)                      !Sngl
118     C      PI    = DACOS(-1.0D0)                   !Dbl
119     C      RAD   = 180.0/PI
120     C      PIO2 = PI/2.0
121     C
122     C.....+.....+.....+.....+.....+...
123     C  \/ TAPE1 must contain trajectory control cards
124     C      Terminate program if no data on TAPE1 file
125     C      Terminate if EOF encountered
126     C      Terminate if negative data found on input file
127     C      Terminate if bad data found on input file
128     C.....+.....+.....+.....+.....+...
129     C
130     100 READ (1,1010,IOSTAT=IOSTAT,ERR=120,END=110) GDLATD,GLOND,PC,
131           *      GDZED,GDAZD,DELPC,INDO,IERRPT,INDEX
132     1010 FORMAT (BZ,6F8.2,3I8)
133     C
134     110 CONTINUE
135     IF (IOSTAT.LT.0) THEN
136       WRITE (*,1020)
137       GO TO 150
138     ENDIF
139     1020 FORMAT (' END OF FILE ON TAPE 1 (DATA INPUT)')
140

```

```

141    120 IF (IOSTAT.GT.0) THEN
142        WRITE (*,1030) IOSTAT,GDLATD,GLOND,PC,DELPC,
143        *           INDO,IERRPT,INDEX
144        GO TO 150
145    ENDIF
146    1030 FORMAT (' ERROR ON DATA INPUT FILE (TAPE1), IOSTAT =',I5/
147        *           4F8.3,3I8)
148    C
149        IF (PC.LE.0) THEN
150            WRITE (*,1040)
151            GO TO 150
152        ENDIF
153    1040 FORMAT (' END OF DATA INPUT (NEGATIVE VALUE READ IN)')
154    C
155        WRITE (*,1050) GDLATD,GLOND,PC,GDZED,GDAZD,DELPC,INDO,IERRPT,INDEX
156    1050 FORMAT (' TAPE 1 ',6F7.2,3I6)
157    C
158        C.....+.....+.....+.....+.....+.....+.....+..
159        C   \/ Start at top of atmosphere (20 km above surface of oblate earth)
160        C   Coding is relic of past when ISALT was read in
161        C.....+.....+.....+.....+.....+.....+..
162        C
163        ISALT = 0
164        IF (ISALT.LE.0) SALT = 20.0
165        IF (ISALT.GT.0) SALT = ISALT
166    C
167        KNT = 0
168        IDELPC = DELPC*1000.0+0.0001
169        INDXPC = PC*1000.0+0.0001
170    C
171        C.....+.....+.....+.....+.....+.....+..
172        C   For trajectories from Earth
173        C   convert from Geodetic coordinates to Geocentric coordinates
174        C   Geodetic coordinates used for input
175        C   GEOCENTRIC coordinates used for output
176        C   coordinates are placed in common block /MNSINGL/
177        C   All calculation are done in Geocentric coordinates!
178        C   \/ Conversion from Geodetic to Geocentric coordinates
179        C.....+.....+.....+.....+.....+..
180        C
181        CALL GDGC (TCD, TSD)
182    C
183        C.....+.....+.....+.....+.....+.....+..
184        C   \/ Remember positron of initial point on trajectory
185        C   in Geocentric coordinates
186    C
187        C   Y(1) is distance in earth radii from geocenter
188        C   Start with height above geoid and convert to earth radii
189        C   The initial values of Y(1), Y(2) AND Y(3) are
190        C   calculated in subroutine GDGC
191    C
192        C   Coordinate reference system
193        C   Y(1) = R      = vertical
194        C   Y(2) = THETA  = south
195        C   Y(3) = PHI    = east
196        C.....+.....+.....+.....+.....+..
197    C
198        RY2 = Y(2)
199        RY3 = Y(3)
200        RY1 = Y(1)
201    C
202        GDAZ = GDAZD/RAD
203        GDZE = GDZED/RAD
204        C   TSGDZE = SIN(GDZE)          !Sngl
205        C   TCGDZE = COS(GDZE)          !Sngl
206        C   TSGDAZ = SIN(GDAZ)          !Sngl
207        C   TCGDAZ = COS(GDAZ)          !Sngl
208        C   TSGDZE = DSIN(GDZE)         !Dbll
209        C   TCGDZE = DCOS(GDZE)         !Dbll
210        C   TSGDAZ = DSIN(GDAZ)         !Dbll

```

```

211      TCGDAZ = DCOS(GDAZ)                               !Dbl
212      C
213      C.....+.....+.....+.....+.....+.....+.....+..
214      C   \/ Get Y1, Y2, Y3 components in Geodetic coordinates
215      C   Azimuth is measured clockwise from the north
216      C   in R, THETA, PHI coordinates, in the THETA-PHI plane
217      C   The angle is 180 - AZD
218      C.....+.....+.....+.....+.....+.....+..+
219      C
220      Y1GD = TCGDZE
221      Y2GD = -TSGDZE*TCGDAZ
222      Y3GD = TSGDZE*TSGDAZ
223      C
224      C.....+.....+.....+.....+.....+.....+..+
225      C   \/ The small angle delta at the point in space between the
226      C   downward Geodetic direction and the
227      C   downward Geocentric direction is given by
228      C   DELTA = Geocentric co-latitude + Geodetic latitude - 90 (deg)
229      C
230      C   We are looking up
231      C   The rotation from Geodetic vertical to Geocentric Vertical
232      C   Is always rotation toward the equator
233      C
234      C   \/ Convert from Geodetic to Geocentric Components for Y1, Y2,
235      C.....+.....+.....+.....+.....+..+
236      C
237      Y1GC = Y1GD*TCD+Y2GD*TSD
238      Y2GC = -Y1GD*TSD+Y2GD*TCD
239      Y3GC = Y3GD
240      C
241      C   WRITE (*,1060) GDZED, GDZE, GDAZD, GDAZ, TSGDZE, TCGDZE, TSGDAZ, TCGDAZ
242      C   WRITE (*,1060) Y1GD, Y2GD, Y3GD, Y1GC, Y2GC, Y3GC
243      C1060 FORMAT (' 1050',8F15.5)
244      C
245      C.....+.....+.....+.....+.....+.....+..+
246      C   ****
247      C   Main control of trajectory calculations begins here
248      C   Trajectories are calculated in subroutine SINGLE
249      C   ****
250      C
251      C   PC      = rigidity IN GV
252      C   INDXPC = index of rigidity in MV (integer)
253      C   IRSLT  = trajectory result
254      C           IRSLT +1    allowed
255      C           IRSLT 0    failed
256      C           IRSLT -1   re-entrant
257      C.....+.....+.....+.....+.....+..+
258      C
259      DO 130 NDO = 1, INDO
260      C
261      IF (IERRPT.GE.1) WRITE (16,1070) GDLATD, GLOND, KNT, INDO, NDO,
262      *                   IDELPC, INDXPC, DELPC, PC
263      C
264      CALL SINGLTJ (PC, IRS LT, INDXPC, Y1GC, Y2GC, Y3GC)
265      C
266      KNT = KNT+1
267      INDXPC = INDXPC-IDELPC
268      PC = FLOAT(INDXPC)/1000.0
269      C
270      C.....+.....+.....+.....+.....+..+
271      C   \/ Check termination conditions
272      C.....+.....+.....+.....+..+
273      C
274      IF (PC .LE. 0.0) GO TO 140
275      IF (TSTEP .GE. FSTEP) GO TO 150
276      C
277      130 CONTINUE
278      140 CONTINUE
279      1070 FORMAT (' 1070 ',2F7.2,5I6,2F6.2)
280      C

```

```

281 C.....+.....+.....+.....+.....+.....+...
282 C ***** End of main control loop *****
283 C
284 C   /\ Go read in next control card
285 C.....+.....+.....+.....+.....+...
286 C
287 C
288 C     GO TO 100
289 C
290 C.....+.....+.....+.....+.....+...
291 C ***** End of trajectory calculations *****
292 C
293 C
294 C.....+.....+.....+.....+.....+...
295 C
296 C     150 CONTINUE
297 C
298 C       WRITE (*, 1120) TSTEP,NTRAJC
299 C       WRITE (8, 1120) TSTEP,NTRAJC
300 C       1120 FORMAT (//' TOTAL NUMBER OF STEPS      ',F15.0//'
301 C                  *          ' TOTAL NUMBER OF TRAJECTORIES',I15//')
302 C       Write (*,1130)
303 C       1130 format (' End program TJALLMAG')
304 C
305 C       STOP
306 C
307 C.....+.....+.....+.....+.....+...
308 C     Y(1) is R coordinate      Y(2) is THETA coordinate
309 C     Y(3) is PHI coordinate    Y(4) is V(R)
310 C     Y(5) is V(THETA)         Y(6) is V(PHI)
311 C     F(1) is R dot           F(2) is THETA dot
312 C     F(3) is PHI dot         F(4) is R dot dot
313 C     F(5) is THETA dot dot  F(6) is PHI dot dot
314 C     BR  is B(R)             BT  is B(THETA)
315 C     BP  is B(PHI)           B   is magnitude of magnetic field
316 C.....+.....+.....+.....+.....+...
317 C
318 C     ierrpt vlu  Program Format Variables printed out
319 C     IERRPT = 1 "MAIN" 1070 Input to SINGLTJ
320 C     IERRPT = 1 SINGLTJ 2000 Input to SINGLTJ
321 C     IERRPT = 2 SINGLTJ 2070 PC,BETA,KBF,RCKBETA,NSTEP,TBETA,Y,H
322 C     IERRPT = 4 SINGLTJ 2090 Y,F,ACCER,H,NSTEP
323 C     IERRPT = 3 SINGLTJ 2100 H,HCK,Y(1),DELACC,PC,NSTEP
324 C     IERRPT = 3 SINGLTJ 2110 H,HCK,Y(1),RFA,    PC,NSTEP
325 C     IERRPT = 3 SINGLTJ 2120 H,HCK,Y(1),NAMX,F(ICK),ICK,FOLD(ICK),
326 C                           ICK,PC,STEP
327 C     IERRPT = 4 SINGLTJ 2130 Y(1),DISCK,PVEL,H,HSNEK,HOLD,NSTEP
328 C     IERRPT = 4 SINGLTJ 2140 Y(1),DISCK,PVEL,H,           HOLD,NSTEP
329 C
330 C     END
331 C     SUBROUTINE GDGC (TCD, TSD)
332 C
333 C.....+.....+.....+.....+.....+...
334 C   /\ Convert from Geodetic to Geocentric coordinates
335 C     Adopted from NASA ALLMAG
336 C     GDLATD = Geodetic latitude (in degrees)
337 C     GCLATD = Geocentric latitude (in degrees)
338 C     GDCLTD = Geodetic co-latitude
339 C     ERPLSQ is earth radius AT poles squared = 40408585 (km sq)
340 C     EREQSQ is earth radius AT equator squared = 40680925 (km sq)
341 C     ERADPR is earth polar radius = 6356.774733 (km)
342 C     ERADER is earth equatorial radius = 6378.160001 (km)
343 C     ERAD  is earth average radius = 6371.25 (km)
344 C     ERADFL is flattening factor = 1.0/298.25
345 C     ERADFL = (ERADEQ - factor)/ERADEQ
346 C     ERECSQ is eccentricity squared = 0.00673966
347 C     ERECSQ = EREQSQ/ERPLSQ - 1.0
348 C.....+.....+.....+.....+...
349 C
350 Clast Mod 15 Jan 97 Common block SNGLR & SNGLI

```

```

351 C      Mod      Feb 96  Standard reference TJ1V line check
352 C
353 C.....+.....+.....+.....+.....+.....+.....+...
354 C      Programmer - Don F. Smart; FORTRAN77
355 C      Note - The programming adheres to the conventional FORTRAN
356 C          default standard that variables beginning with
357 C          'i','j','k','l','m', or 'n' are integer variables
358 C          Variables beginning with "c" are character variables
359 C          All other variables are real
360 C.....+.....+.....+.....+.....+.....+...
361 C          Do not mix different type variables in same common block
362 C          Some computers do not allow this
363 C.....+.....+.....+.....+.....+...
364 C
365 C
366      IMPLICIT INTEGER (I-N)
367      IMPLICIT REAL * 8(A-B)
368      IMPLICIT REAL * 8(D-H)
369      IMPLICIT REAL * 8(O-Z)
370 C
371 C.....+.....+.....+.....+.....+.....+...
372 C
373      COMMON /WRKVLU/ F(6),Y(6),ERAD,EOMC,VEL,BR,BT,BP,B
374      COMMON /WRKTSC/ TSY2,TCY2,TSY3,TCY3
375      COMMON /TRIG/   PI,RAD,PIO2
376      COMMON /GEOID/  ERADPL, ERECSQ
377      COMMON /SNGLR/  SALT,DISOUT,GCLATD,GDLATD,GLOND,GDAZD,GDZED,
378      *                  RY1,RY2,RY3,RHT,TSTEP
379 C
380 C.....+.....+.....+.....+.....+...
381 C
382      ERPLSQ = 40408585.0
383      EREQSQ = 40680925.0
384      C      ERADPL = SQRT(ERPLSQ)                                !Sngl
385      C      ERADPL = DSQRT(ERPLSQ)                               !Db1l
386      C      ERECSQ = EREQSQ/ERPLSQ - 1.0
387 C
388      GDCLT = PIO2-GDLATD/RAD
389      C      TSGDCLT = SIN(GDCLT)                                !Sngl
390      C      TCGDCLT = COS(GDCLT)                                !Sngl
391      C      TSGDCLT = DSIN(GDCLT)                               !Db1l
392      C      TCGDCLT = DCOS(GDCLT)                               !Db1l
393
394      ONE = EREQSQ*TSGDCLT*TSGDCLT
395      TWO = ERPLSQ*TCGDCLT*TCGDCLT
396      THREE = ONE+TWO
397      C      RHO = SQRT(THREE)                                !Sngl
398      C      RHO = DSQRT(THREE)                               !Db1l
399 C
400 C.....+.....+.....+.....+.....+...
401 C      \/ Get geocentric distance from geocenter in kilometers
402 C.....+.....+.....+.....+...
403 C
404      C      DISTKM = SQRT(SALT*(SALT+2.0*RHO)+(EREQSQ*ONE+ERPLSQ*TWO)/THREE) !Sngl
405      C      DISTKM = DSQRT(SALT*(SALT+2.0*RHO)+(EREQSQ*ONE+ERPLSQ*TWO)/THREE) !Db1l
406 C
407 C.....+.....+.....+.....+.....+...
408 C      TCD and TSD are sine and cosine of the angle the Geodetic vertical
409 C          must be rotated to form the Geocentric Vertical
410 C.....+.....+.....+.....+...
411 C
412      C      TCD = (SALT+RHO)/DISTKM
413      C      TSD = (EREQSQ-ERPLSQ)/RHO*TCGDCLT*TSGDCLT/DISTKM
414      C      TCY2 = TCGDCLT*TCD-TSGDCLT*TSD
415      C      TSY2 = TSGDCLT*TCD+TCGDCLT*TSD
416 C
417      C      Y(2) = ACOS(TCY2)                                !Sngl
418      C      Y(2) = DACOS(TCY2)                               !Db1l
419      C      Y(3) = GLOND/RAD
420      C      Y(1) = DISTKM/ERAD

```

```

421 C
422      GCLATD = (PIO2-Y(2))*RAD
423 C
424 C      WRITE (*,1200) GDLATD,GDCLT,TSGDCLT,TCGDCLT,ONE,TWO,THREE,RHO
425 C1200 FORMAT (' 1200',8F15.5)
426 C      WRITE (*,1200) DISTKM,TCD,TSD,TCY2,TSY2,GCLATD
427 C
428      RETURN
429 END
430 SUBROUTINE SINGLTJ (PC,IRSLT,INDXPC,Y1GC,Y2GC,Y3GC)
431 C
432 C.....+.....+.....+.....+.....+.....+.....+...
433 C      Cosmic-ray trajectory calculations subroutine
434 C          calculates cosmic ray trajectory at rigidity PC
435 C.....+.....+.....+.....+.....+.....+.....+...
436 C          PC = rigidity in GV
437 C          IRSLT = trajectory result
438 C          INDXPC = index of rigidity in mv (integer)
439 C              Y1GC,Y2GC,Y3GC are initial geocentric coordinates
440 C.....+.....+.....+.....+.....+.....+...
441 C      \/ step size optimization & look ahead for potential BETA problems
442 C          monitor accelerating terms and reduce step length
443 C              if large increase occurs
444 C          Restart at smaller step size if BETA error occurs
445 C.....+.....+.....+.....+.....+.....+...
446 C          Restrictions: Cannot run over N or S pole; will get BETA blowup
447 C.....+.....+.....+.....+.....+.....+...
448 CLast Mod 17 Feb 99 if (ymax.lt.6.6) IFATE = 3
449 C Mod 18 Jan 97 Patch high latitude beta problem
450 C Mod Jan 97 High latitude step size adjust, introduce AHLT
451 C Mod Jun 96 EDIF limit set to 1.0e-5
452 C Mod Jun 96 IERRPT formats, Boundary and look ahead
453 C Mod FEB 96 standard reference TJ1V (line check 17 Feb)
454
455 C.....+.....+.....+.....+.....+.....+...
456 C      Programmer - Don F. Smart; FORTRAN77
457 C      Note - The programming adheres to the conventional FORTRAN
458 C          default standard that variables beginning with
459 C          'i','j','k','l','m',or 'n' are integer variables
460 C          Variables beginning with "c" are character variables
461 C          All other variables are real
462 C.....+.....+.....+.....+.....+...
463 C          Do not mix different type variables in same common block
464 C          Some computers do not allow this
465 C.....+.....+.....+.....+.....+...
466 C
467 IMPLICIT INTEGER (I-N)
468 IMPLICIT REAL * 8(A-B)
469 IMPLICIT REAL * 8(D-H)
470 IMPLICIT REAL * 8(O-Z)
471 C
472 C.....+.....+.....+.....+.....+...
473 C
474 COMMON /WRKVLU/ F(6),Y(6),ERAD,EOMC,VEL,BR,BT,BP,B
475 COMMON /WRKTSC/ TSY2,TCY2,TSY3,TCY3
476 COMMON /TRIG/ PI,RAD,PIO2
477 COMMON /GEOID/ ERADPL, ERECSQ
478 COMMON /SNGLR/ SALT,DISOUT,GCLATD,GDLATD,GLOND,GAZD,GDZED,
479 *                  RY1,RY2,RY3,RHT,TSTEP
480 COMMON /SNGLI/ LIMIT,NTRAJC,IERRPT
481 C
482 C.....+.....+.....+.....+.....+...
483 C
484 DIMENSION P(6),Q(6),R(6),S(6),YB(6),FOLD(6),YOLD(6)
485 C
486 C.....+.....+.....+.....+.....+...
487 C
488 CHARACTER*1 CF,CR
489 CHARACTER*6 CNAME
490 C

```

```

491      DATA CF,CR / 'F','R'/                                #####
492      DATA CNAME / ' I95 '/
493      C
494      C.....+.....+.....+.....+.....+.....+...
495      C
496      IF (IERRPT.GT.0) WRITE (16,2000) PC,IDXPC,RY1,RY2,RY3
497      2000 FORMAT (' SINGLTJ ',F8.3,I8,3F8.4)
498      C
499      BETAST = 2.0
500      LSTEP = 0
501      KBF = 0
502      C
503      C.....+.....+.....+.....+.....+.....+...
504      C  \/ Runge-Kutta constants
505      C.....+.....+.....+.....+.....+.....+...
506      C
507      RC1O6 = 1.0D0/6.0D0                               !Dbl
508      C  SR2 = SQRT(2.0)                                !Sngl
509      C  SR2 = DSQRT(2.0D0)                            !Dbl
510      C  TMS2O2 = (2.0-SR2)/2.0
511      C  TPS2O2 = (2.0+SR2)/2.0
512      C
513      C.....+.....+.....+.....+.....+...
514      C  \/ Initialize Runge-Kutta variables to zero
515      C.....+.....+.....+.....+.....+...
516      C
517      100 DO 110 I = 1, 6
518      YB(I) = 0.0
519      S(I) = 0.0
520      R(I) = 0.0
521      Q(I) = 0.0
522      P(I) = 0.0
523      F(I) = 0.0
524      110 CONTINUE
525      C
526      NMAX = 0
527      NMIN = 0
528      NSTEP = 0
529      NSTEPT = 0
530      C
531      TAU = 0.0
532      TU100 = 0.0
533      YMAX = RY1
534      C
535      C.....+.....+.....+.....+.....+...
536      C  \/ Define initial point at start of trajectory
537      C  Y(1), Y(2), Y(3) are the position vectors
538      C.....+.....+.....+.....+...
539      C
540      Y(1) = RY1
541      Y(2) = RY2
542      Y(3) = RY3
543      C  GRNDKM = (ERADPL/SQRT(1.0-ERECSQ*TSY2SQ))          !Sngl
544      C  GRNDKM = (ERADPL/DSQRT(1.0-ERECSQ*TSY2SQ))        !Dbl
545      C  Y10 = (RHT+GRNDKM)/ERAD
546      C  R120KM = (ERAD+120.0)/ERAD
547      C
548      C.....+.....+.....+.....+...
549      C  Rigidity = momentum/charge
550      C  use oxygen 16 as reference isotope
551      C  Constants used from Handbook of Physics (7-170)
552      C  1 amu = 0.931141  GeV
553      C.....+.....+.....+.....+...
554      C
555      ANUC = 16.0
556      ZCHARGE = 8.0
557      C
558      EMCSQ = 0.931141
559      C  TENG = SQRT((PC*ZCHARGE)**2+(ANUC*EMCSQ)**2)      !Sngl
560      C  TENG = DSQRT((PC*ZCHARGE)**2+(ANUC*EMCSQ)**2)    !Dbl

```

```

561      EOMC = -8987.566297*ZCHARGE/TENG
562      C      GMA = SQRT(((PC*ZCHARGE)/(EMCSQ*ANUC))**2+1.0)          !Sngl
563      C      BETA = SQRT(1.0-1.0/(GMA*GMA))                         !Sngl
564      C      GMA = DSQRT(((PC*ZCHARGE)/(EMCSQ*ANUC))**2 + 1.0D0)       !Dbl
565      C      BETA = DSQRT(1.0D0 - 1.0D0/(GMA*GMA))                      !Dbl
566      PVEL = VEL*BETA
567      HMAX = 1.0/PVEL
568      DISCK = DISOUT - 1.1*HMAX*PVEL
569      C
570      C.....+.....+.....+.....+.....+.....+.....+...
571      C      \/ Set max step length ("HMAX") to 1 earth radii
572      C      PVEL is particle velocity in earth radii per second
573      C      DISCK is check for approaching termination boundary
574      C          (within 1.1 steps)
575      C.....+.....+.....+.....+.....+.....+...
576      C
577      EDIF = BETA*1.0E-4
578      IF (EDIF.LT.1.0-5) EDIF = 1.0E-5
579      IF (BETA.LT.0.1)   EDIF = 1.0E-4
580      C
581      C.....+.....+.....+.....+.....+.....+...
582      C      \/ Y(4), Y(5), Y(6) are the velocity vectors
583      C.....+.....+.....+.....+.....+...
584      C
585      Y(4) = BETA*Y1GC
586      Y(5) = BETA*Y2GC
587      Y(6) = BETA*Y3GC
588      C
589      AZD = GDAZD
590      ZED = GDZED
591      IAZ = AZD+0.01
592      IZE = ZED+0.01
593      C
594      C.....+.....+.....+.....+.....+.....+...
595      C      \/ Set HSTART to about 1 % of the time to complete one gyro-radius
596      C          in a 1 Gauss field
597      C      H = [(2.0*PI*33.333*PC)/(BETA*C)]/0.01
598      C          if restart after BETA error, set HCK to small value
599      C          Introduce AHLT to control step size at high lat (beta problem)
600      C          HCK - reduce step size when large acceleration
601      C          HOLD - last step size used
602      C          HCNG - only allow 20% max growth in step size
603      C          HSNEK - attempt to approach boundary quickly
604      C          Problem at z=90 at high lat
605      C          add zen angle in deg to reduce first step
606      C.....+.....+.....+.....+.....+...
607      C
608      PTCY2 = ABS(TCY2)
609      AHLT = (1.0 + PTCY2)**2
610      HSTART = 6.0E-6*PC/(BETA*AHLT + ZED*PTCY2)
611      IF (HSTART.LT.1.0E-6) HSTART = 1.0E-6
612      HOLD = HSTART
613      HCK = HSTART
614      HCNG = HSTART
615      C
616      C      WRITE (16, 2010) HMAX,HOLD,HCK,HCNG,Y(4),Y(5),Y(6),PVEL, NSTEP
617      C2010 FORMAT (' 2010 ',18X, 4F9.6, 3F9.4, F9.4,9X,15X,I6)
618      C
619      C.....+.....+.....+.....+.....+...
620      C      Start Runge-Kutta
621      C      \/\ /\ /\ /\ /\ /\ /
622      C      \/\ /\ /\ /\ /\ /
623      C      \/\ /\ /\ /
624      C      \/\ /
625      C      \
626      C.....+.....+.....+.....+.....+...
627      C      Change in step size criteria, Aug 97
628      C      remove cos VxB step size, causes problems in tight loops
629      C      step size is now only a function of B and BETA
630      C.....+.....+.....+.....+...

```

```

631   C
632   130 IF (HCK.LT.1.0E-6) HCK = 1.0E-6
633       CALL FGRADA
634       HB = 1.6E-5*PC/(B*BETA)
635       H = HB/BETAST
636   C
637       IF (KBF.GT.0) H=H/(FLOAT(KBF*2))
638       IF (H.GT.HMAX) H = HMAX
639       IF (H.GT.HCNG) H = HCNG
640       IF (H.GT.HCK) H = HCK
641   C
642       DO 140 I = 1, 6
643           S(I) = H*F(I)
644           P(I) = 0.5*S(I)-Q(I)
645           YB(I) = Y(I)
646           Y(I) = Y(I)+P(I)
647           R(I) = Y(I)-YB(I)
648           Q(I) = Q(I)+3.0*R(I)-0.5*S(I)
649   140 CONTINUE
650   C
651       CALL FGRADA
652   C
653       DO 150 I = 1, 6
654           S(I) = H*F(I)
655           P(I) = TMS2O2*(S(I)-Q(I))
656           YB(I) = Y(I)
657           Y(I) = Y(I)+P(I)
658           R(I) = Y(I)-YB(I)
659           Q(I) = Q(I)+3.0*R(I)-TMS2O2*S(I)
660   150 CONTINUE
661   C
662       CALL FGRADA
663   C
664       DO 160 I = 1, 6
665           S(I) = H*F(I)
666           P(I) = TPS2O2*(S(I)-Q(I))
667           YB(I) = Y(I)
668           Y(I) = Y(I)+P(I)
669           R(I) = Y(I)-YB(I)
670           Q(I) = Q(I)+3.0*R(I)-TPS2O2*S(I)
671   160 CONTINUE
672   C
673       CALL FGRADA
674   C
675       DO 170 I = 1, 6
676           S(I) = H*F(I)
677           P(I) = RC1O6*(S(I)-2.0*Q(I))
678           YB(I) = Y(I)
679           Y(I) = Y(I)+P(I)
680           R(I) = Y(I)-YB(I)
681           Q(I) = Q(I)+3.0*R(I)-0.5*S(I)
682   170 CONTINUE
683   C
684   C.....+.....+.....+.....+.....+.....+...
685   C      /\
686   C      /\ \
687   C      /\ /\ \
688   C      /\ /\ /\ /\ \
689   C      /\ /\ /\ /\ /\ \
690   C      One Runge-Kutta
691   C      step completed
692   C.....+.....+.....+.....+.....+.....+...
693   C
694   NSTEP = NSTEP+1
695   NSTEPT = NSTEPT + 1
696   TAU = TAU+H
697   HOLD = H
698   HCNG = H*1.2
699   HCK = HCNG
700   C

```

```

701 C.....+.....+.....+.....+.....+.....+...
702 C   \/ Emergency diagnostic printout if desired
703 C.....+.....+.....+.....+.....+.....+...
704 C   WRITE (16, 2030)      H, Y(1),Y(2),Y(3),    PVEL,B, NSTEP
705 C   WRITE (16, 2040)  HB,H,HMAX,HOLD,HCK,HCNG,Y(4),Y(5),Y(6),
706 C   *                  PVEL,B,NSTEP
707 C2030 FORMAT (' 2030 ', 9X,     F9.6, 36X,     3F9.5,F9.4,F9.5,18X,I6)
708 C2040 FORMAT (' 2040 ',           6F9.6,            3F9.5,F9.4,F9.5,18X,I6)
709 C
710 C.....+.....+.....+.....+.....+.....+...
711 C   \/ Check for altitude less than 100 km
712 C       if less than 120 km, compute exact altitude above oblate earth
713 C       and sum time trajectory is below 100 km altitude.
714 C       set re-entrant altitude at RHT km above oblate earth
715 C           computed from international reference ellipsoid
716 C.....+.....+.....+.....+.....+.....+...
717 C
718 C   IF (Y(1).LT.R120KM) THEN
719 C       TSY2SQ = SIN(Y(2))**2                                !Sngl
720 C       TSY2SQ = DSIN(Y(2))**2                               !Dbl
721 C       GRNDKM = (ERADPL/SQRT(1.0-ERECSQ*TSY2SQ))          !Sngl
722 C       GRNDKM = (ERADPL/DSQRT(1.0-ERECSQ*TSY2SQ))         !Dbl
723 C       R100KM = (100.0+GRNDKM)/ERAD
724 C       R120KM = (120.0+GRNDKM)/ERAD
725 C       IF (Y(1).LT.R100KM) TU100 = TU100+H
726 C       PSALT = Y(1)*ERAD-GRNDKM
727 C       Y10 = (RHT+GRNDKM)/ERAD
728 C
729 C   IF (NSTEP.GT.5) THEN
730 C       IF (Y(1).LT.Y10.OR.PSALT.LE.0.0) THEN
731 C           IF (IERRPT.GT.2) WRITE (16, 2045) PSALT, Y(1), Y10
732 C           IRT = -1
733 C           GO TO 260
734 C   ENDIF
735 C   ENDF
736 C
737 C2045 FORMAT (' 2045 PSALT,Y(1),Y10',F10.6,1PE14.6,E14.6)
738 C
739 C.....+.....+.....+.....+.....+...
740 C   \/ Begin error checks
741 C       (1) Check for unacceptable changes in BETA
742 C.....+.....+.....+.....+.....+...
743 C
744 C   RCKBETA = SQRT(Y(4)*Y(4)+Y(5)*Y(5)+Y(6)*Y(6))          !Sngl
745 C   RCKBETA = DSQRT(Y(4)*Y(4)+Y(5)*Y(5)+Y(6)*Y(6))        !Dbl
746 C   TBETA = BETA-RCKBETA
747 C   IF (ABS(TBETA).GT.EDIF) THEN
748 C       KBF = KBF+1
749 C       BETAST = BETAST + AHLT
750 C       EDIF = 2.0*EDIF
751 C       IF (RCKBETA.GT.(1.0+EDIF)) THEN
752 C           BETAST = BETAST+FLOAT(KBF)*(1.0+AHLT)
753 C           WRITE (*,2050) KBF,BETA,RCKBETA
754 C           WRITE (*,2060) Y,H,PC,NSTEP
755 C           WRITE (16,2050) KBF,BETA,RCKBETA
756 C           WRITE (16,2060) Y,H,PC,NSTEP
757 C   ENDIF
758 C
759 C   WRITE (16,2070) PC,BETA,KBF,RCKBETA,NSTEP,TBETA,Y,H
760 C   WRITE (*,2070) PC,BETA,KBF,RCKBETA,NSTEP,TBETA,Y,H
761 C
762 C.....+.....+.....+.....+.....+...
763 C   \/ Check for irrecoverable beta error
764 C       if KBF > 4, set fate to failed and start next rigidity
765 C.....+.....+.....+.....+.....+...
766 C
767 C   IF (KBF.lt.5) THEN
768 C       GO TO 100
769 C   ELSE
770 C       IRT = 0

```

```

771      PATH = -PVEL*TAU
772      ISALT = SALT+0.0001
773      WRITE (*,2080)
774      GO TO 280
775      ENDIF
776      ENDIF
777 C
778 2050 FORMAT (' 2050 ','KBF=',i5,5x,' error, the velocity of the ',
779      *      'particle (BETA) has exceeded the velocity of light'
780      *      ' BETA at start of trajectory was ',F10.7/
781      *      ' BETA now equals           ',F10.7/
782      *      ' reduce step size and try again   ')
783 2060 FORMAT (' 2060 ','Y,H,PC,NSTEP=',8F12.6,I10)
784 2070 FORMAT (' 2070 ','ERROR, Particle BETA changed;',
785      *      ' PC = ',F10.6,' GV'
786      *      ' 6x, ' Beta at start was ',F10.7,17X,'KBF=',I6/
787      *      ' 6x, ' Beta new equals   ',F10.7,10X,'step number',I6/
788      *      ' 6x, ' Beta difference  ',F10.7/
789      *      ' 6x, ' step length reduced and trajectory recalculated ''
790      *      ' 6x, 'Y=',6F12.6,6X,' H=',F15.7)
791 2080 FORMAT (' 2080 ','Irrecoverable BETA error problem'/6x,
792      *      ' Set fate to Failed & make path length negative'/6x,
793      *      ' Terminate this trajectory & continue with program')
794 C
795 C     ...+.....+.....+.....+.....+.....+.....+...
796 C     \/ Continue status checks
797 C     (2) Compute composite acceleration
798 C     ...+.....+.....+.....+.....+.....+...
799 C
800 C     ACCER = SQRT(F(4)*F(4)+F(5)*F(5)+F(6)*F(6))          !Sngl
801 C     ACCER = DSQRT(F(4)*F(4)+F(5)*F(5)+F(6)*F(6))          !Dbl
802 C
803 C     IF (IERRPT.GT.3) WRITE (16,2090) Y, F, ACCER, H, NSTEP
804 2090 FORMAT (' Y,F,A,H ',f7.4,5f7.3,1x,1pe8.1,5e8.1,1x,e8.1, e9.2,I6)
805 C
806 C     ...+.....+.....+.....+.....+.....+...
807 C     \/ Continue status checks, make adjustment latitude dependant
808 C     (3) Monitor change in composite acceleration
809 C         If composite acceleration (new-old) change > 5
810 C         If composite acceleration (new/old) ratio > 2
811 C             change step size to a smaller value
812 C     ...+.....+.....+.....+.....+...
813 C
814 C     IF (NSTEP.GE.2) THEN
815 C         IF (ACCER.GT.ACcold) THEN
816 C             DELACC = ACCER-ACcold
817 C             IF (DELACC.GT.5.0) THEN
818 C                 HCK = HCK/(1.0+AHLT)
819 C                 IF (IERRPT.GT.2) WRITE (16,2100)
820 C                     H,HCK,y(1),DELACC,PC,NSTEP
821 C                     RFA = ACCER/ACcold
822 C                     IF (RFA.GT.2.0) THEN
823 C                         HCK = HCK/(1.0+AHILT)
824 C                         IF (IERRPT.GT.2) WRITE (16,2110)
825 C                             H,HCK,Y(1),RFA,PC,NSTEP
826 C             ENDIF
827 C         ENDIF
828 C     ENDIF
829 C
830 2100 FORMAT (' 2100 ','H-REDUCE',2x,'H=',F8.6,2x,'HCK=',F8.6,2x,
831      *      'Y(1)=',f7.4,2X,'DELACC=',F6.2,4X,'PC=',F8.3,4x,'NSTEP=',I8)
832 2110 FORMAT (' 2110 ','H-REDUCE',2x,'H=',F8.6,2x,'HCK=',F8.6,2x,
833      *      'Y(1)=',f7.4,4X,' RFA=',F6.2,4X,'PC=',F8.3,4x,'NSTEP=',I8)
834 C
835 C     ...+.....+.....+.....+.....+...
836 C     \/ Continue status checks, make adjustment latitude dependant
837 C     (4) Monitor change in acceleration components
838 C         If change in any acceleration component is more than
839 C             a factor of 3, reduce step length
840 C     ...+.....+.....+.....+...

```

```

841   C
842       DO 200 ICK = 4, 6
843           AFOLD = ABS(FOLD(ICK))
844           IF (AFOLD.GT.3.0) THEN
845               RFCK = ABS(F(ICK)/AFOLD)
846               IF (RFCK.GT.3.0) THEN
847                   HCK = HCK/(1.0+AHLT)
848                   IF (IERRPT.GT.2) THEN
849                       WRITE (16,2120) H, HCK, Y(1), NMAX, ICK, F(ICK),
850                           ICK, FOLD(ICK), PC, NSTEP
850       &
851           ENDIF
852       ENDIF
853   ENDIF
854   200   CONTINUE
855   ENDIF
856   C
857   2120 FORMAT (' 2120 ', 'H-reduce', 2X, 'H=' ,F8.6, 2X, 'HCK=' ,F8.6, 2X,
858   *          'Y(1)' ,F7.4, 2X, 'NAMX=' ,I4, 2X, 'F(' ,I1, ')' =', F6.2, 2X,
859   *          'FOLD(' ,I1, ')' =', F6.2, 2X, 'PC=' ,F6.3, 2X, 'NSTEP=' ,I6)
860   C
861       ACCOLD = ACCER
862   C.....+.....+.....+.....+.....+.....+.....+..
863   C /\ Error checks complete
864   C.....+.....+.....+.....+.....+.....+..
865   C.....+.....+.....+.....+.....+.....+..
866   C.....+.....+.....+.....+.....+.....+..
867   C.....+.....+.....+.....+.....+.....+..
868   C.....+.....+.....+.....+.....+..
869   C.....+.....+.....+.....+.....+..
870   C.....+.....+.....+.....+.....+..
871   C.....+.....+.....+.....+.....+..
872   C.....+.....+.....+.....+..
873   C.....+.....+.....+.....+..
874   C.....+.....+.....+.....+..
875   C.....+.....+.....+.....+..
876   C.....+.....+.....+.....+..
877   C.....+.....+.....+.....+..
878   C.....+.....+.....+..
879   C.....+.....+.....+..
880   C.....+.....+.....+..
881   C.....+.....+.....+..
882   C.....+.....+.....+..
883   C.....+.....+.....+..
884   C.....+.....+.....+..
885   C.....+.....+.....+..
886   C.....+.....+.....+..
887   C.....+.....+.....+..
888   C.....+.....+.....+..
889   C.....+.....+.....+..
890   C.....+.....+.....+..
891   C.....+.....+.....+..
892   C.....+.....+.....+..
893   C.....+.....+.....+..
894   C.....+.....+.....+..
895   C.....+.....+.....+..
896   C.....+.....+.....+..
897   C.....+.....+.....+..
898   C.....+.....+.....+..
899   C.....+.....+.....+..
900   C.....+.....+.....+..
901   C.....+.....+.....+..
902   C.....+.....+.....+..
903   C.....+.....+.....+..
904   C.....+.....+.....+..
905   C.....+.....+.....+..
906   C.....+.....+.....+..
907   C.....+.....+.....+..
908   C.....+.....+.....+..
909   C.....+.....+.....+..
910   C.....+.....+.....+..

```

```

911          HCK = 1.0E-5
912          HCNG = 1.0E-5
913      ENDIF
914      C      IF (IERRPT.GT.3) WRITE (16,2130) Y(1),DISCK,PVEL,H,HSNEK,NSTEP
915
916      C      210  IF (Y(1).GT.DISOUT) THEN
917          IF (H.LE.1.0E-5) THEN
918              IRT = 1
919              GO TO 260
920          ENDIF
921          TAU = TAU - H
922
923      C      .....+.....+.....+.....+.....+...
924      C      \/ Backup option invoked if you are here
925      C      .....+.....+.....+.....+.....+...
926      C
927      C      DO 220 I = 1, 6
928          Y(I) = YOLD(I)
929          F(I) = FOLD(I)
930      220  CONTINUE
931      ENDIF
932      GO TO 130
933
934      ENDIF
935      2130 FORMAT (' 2130 ',2X,'Y(1),DISCK,PVEL,H,HSNEK',
936      *           4X,1PE12.6,4X,E12.6,4X,E12.6,4X,2E9.2,22X,I6)
937
938      C      +.....+.....+.....+.....+.....+...
939      C      \/ Have penetrated boundary if you are here.
940      C      if large step size, go back one step and
941      C      reduce step length (and adjust "TAU")
942      C      +.....+.....+.....+.....+...
943
944      C      230  IF (Y(1).GT.DISOUT) THEN
945      C          IF (IERRPT.GT.3) WRITE (16, 2140) Y(1),DISCK,PVEL,H,NSTEP
946
947      C          IF (H.LT.1.0E-5 .OR. HCK.LT.1.0E-5 .OR. HCNG.LT.1.0E-5) THEN
948          IRT = 1
949          GO TO 260
950
951      ELSE
952          HCK = HCK/2.0
953          HCNG = HCNG/2.0
954          TAU = TAU - H
955          DO 240 I = 1, 6
956              Y(I) = YOLD(I)
957              F(I) = FOLD(I)
958      240  CONTINUE
959      GO TO 130
960
961      ENDIF
962
963      C      2140 FORMAT (' 2140 ',2X,'Y(1),DISCK,PVEL,H',
964      *           4X,1PE12.6,4X,E12.6,4X,E12.6,4X,E9.2,27X,I6)
965
966      C      .....+.....+.....+.....+.....+...
967      C      \/ STORE VALUES OF Y AND F AS FOLD & YOLD
968      C      .....+.....+.....+.....+.....+...
969
970      C      DO 250 I = 1, 6
971          YOLD(I) = Y(I)
972          FOLD(I) = F(I)
973      250 CONTINUE
974
975      C      GO TO 130
976
977      C      .....+.....+.....+.....+.....+...
978      C***** ****
979      C      ***** ***** ***** ***** ****
980      C      ***** *****
```

```

981 C      ****
982 C      TRAJECTORY COMPLETE IF YOU ARE HERE
983 C.....+.....+.....+.....+.....+.....+...
984 C
985 260 CONTINUE
986 C
987     IF (Y(1).GE.DISOUT) IRT = 1
988     PATH = PVEL*TAU
989     ISALT = SALT+0.0001
990     LSTEP = BETAST - 1.9
991 C
992 C.....+.....+.....+.....+.....+.....+...
993 C      \/ WRITE OUT RESULTS
994 C      IRT    +1    ALLOWED      (FATE = 0)
995 C      IRT    0     FAILED       (FATE = 2)
996 C      IRT   -1    RE-ENTRANT  (FATE = 1)
997 C.....+.....+.....+.....+.....+...
998 C
999     IF (IRT.GT.0) THEN
1000    C      TCY2 = COS(Y(2))          !Sngl
1001    C      TSY2 = SIN(Y(2))          !Sngl
1002    C      TCY2 = DCOS(Y(2))        !Dbll
1003    C      TSY2 = DSIN(Y(2))        !Dbll
1004    C      YDA5 = Y(5)*TCY2+Y(4)*TSY2
1005    C      ATRG1 = Y(4)*TCY2-Y(5)*TSY2
1006    C      ATRG2 = SQRT(Y(6)*Y(6)+YDA5*YDA5) !Sngl
1007    C      ATRG2 = DSQRT(Y(6)*Y(6)+YDA5*YDA5) !Dbll
1008    C      FASLAT = 0.0
1009    C      IF (ATRG1.NE.0.0.AND.ATRG2.NE.0.0)
1010    C      *      FASLAT = ATAN2(ATRG1,ATRG2)*RAD !Sngl
1011    C      IF (ATRG1.NE.0.0.AND.ATRG2.NE.0.0)
1012    C      *      FASLAT = DATAN2(ATRG1,ATRG2)*RAD !Dbll
1013    C      FASLON = Y(3)*RAD
1014    C      IF (Y(6).NE.0.0.AND.YDA5.NE.0.0)
1015    C      *      FASLON = (Y(3)+ATAN2(Y(6),YDA5))*RAD !Sngl
1016    C      IF (Y(6).NE.0.0.AND.YDA5.NE.0.0)
1017    C      *      FASLON = (Y(3)+DATAN2(Y(6),YDA5))*RAD !Db1
1018    C      IF (FASLON.LT.0.0) FASLON = FASLON+360.0
1019    C      IF (FASLON.GT.360.0) FASLON = FASLON-360.0
1020    C
1021    C      WRITE (8,2150) GDLATD,GCLATD,GLOND,IZE,IAZ,PC,FASLAT,FASLON,
1022    C      *          PATH,NMAX,NSTEP,TU100,YMAX,LSTEP,SALT,CNAME
1023    C
1024    C      IFATE = 0
1025    C      WRITE (7,2160) GDLATD,GLOND,PC,ZED,AZD,ISALT,FASLAT,FASLON,
1026    C      *          NSTEP,IFATE,CNAME
1027    C      ENDIF
1028    2150 FORMAT (2F7.2,F9.2,I5,I4,F10.3,2F8.2,F11.5,I4,I7,F9.5,F9.4,
1029    C      *          I4,F11.1,1X,A6,13X)
1030    2160 FORMAT (F7.2,F8.2,F9.3,2F6.1,I7,F7.2,F8.2,I7,3X,I3,3X,A6)
1031    C
1032    C      IF (IRT.LT.0) THEN
1033    C          RENLAT = (PIO2-Y(2))*RAD
1034    C          RENLON = Y(3)*RAD
1035    C
1036    C      WRITE (8,2170) GDLATD,GCLATD,GLOND,IZE,IAZ,PC,CR,CR,PATH,NMAX
1037    C      *          ,NSTEP,TU100,YMAX,LSTEP,SALT,CNAME,RENLAT,RENLON
1038    C
1039    C      IFATE = 1
1040    C      WRITE (7,2180) GDLATD,GLOND,PC,ZED,AZD,ISALT,NSTEP,IFATE,CNAME
1041    C      ENDIF
1042    2170 FORMAT (2F7.2,F9.2,I5,I4,F10.3,5X,A1,2X,5X,A1,2X,F11.5,I4,I7,
1043    C      *          F9.5,F9.4,I4,F11.1,1X,A6,F6.1,F7.1)
1044    2180 FORMAT (F7.2,F8.2,F9.3,2F6.1,I7,4X,'R',7X,'R',I9,3X,I3,3X,A6)
1045    C
1046    C      280 IF (IRT.EQ.0) THEN
1047    C
1048    C          WRITE (8,2190) GDLATD,GCLATD,GLOND,IZE,IAZ,PC,CF,CF,PATH,
1049    C          *          NMAX,NSTEP,TU100,YMAX,LSTEP,SALT,CNAME
1050    C

```

```

1051      IFATE = 2
1052      IF (YMAX.LT.6.6) IFATE = 3
1053      WRITE (7,2200) GDLATD,GLOND,PC,ZED,AZD,ISALT,NSTEP,IFATE,CNAME
1054      ENDIF
1055      2190 FORMAT (2F7.2,F9.2,I5,I4,F10.3,5X,A1,2X,5X,A1,2X,F11.5,I4,I7,
1056      *          F9.5,F9.5,I4,F11.1,1X,A6,13X)
1057      2200 FORMAT (F7.2,F8.2,F9.3,2F6.1,I7,4X,'F',7X,'F',I9,3X,I3,3X,A6)
1058      C
1059      NTRAJC = NTRAJC+1
1060      TSTEP = TSTEP+FLOAT(NSTEP)
1061      C
1062      C      ...+.....+.....+.....+.....+.....+.....+...
1063      C      \/ Comment out to reduce IO
1064      C      ...+.....+.....+.....+.....+.....+...
1065      C
1066      C      WRITE (*,2210) PC, ZED, AZD, NSTEP, IFATE
1067      C2210 FORMAT (1H+, 22X, 3F7.2,7x,2I6)
1068      C
1069      IRSLT = IRT
1070      RETURN
1071      END
1072      SUBROUTINE FGRADA
1073      C
1074      C.....+.....+.....+.....+.....+.....+.....+...
1075      C      \/ Version of FGRAD to run with NASA NSSDC ALLMAG subroutine
1076      C.....+.....+.....+.....+.....+.....+.....+...
1077      Clast Mod 22 Dec 00 Use NASA ALLMAG for magnetic field calculations
1078      C      Mod Feb 96 standard reference TJ1V (line check 17 Feb)
1079      C.....+.....+.....+.....+.....+.....+...
1080      C      Programmer - Don F. Smart; FORTRAN77
1081      C      Note - The programming adheres to the conventional FORTRAN
1082      C      default standard that variables beginning with
1083      C      'i','j','k','l','m', or 'n' are integer variables
1084      C      Variables beginning with "c" are character variables
1085      C      All other variables are real
1086      C.....+.....+.....+.....+.....+...
1087      C      Do not mix different type variables in same common block
1088      C      Some computers do not allow this
1089      C.....+.....+.....+.....+...
1090      C
1091      IMPLICIT INTEGER (I-N)
1092      IMPLICIT REAL * 8(A-B)
1093      IMPLICIT REAL * 8(D-H)
1094      IMPLICIT REAL * 8(O-Z)
1095      C
1096      C.....+.....+.....+.....+...
1097      C
1098      COMMON /WRKVLU/ F(6),Y(6),ERAD,EOMC,VEL,BR,BT,BP,B
1099      COMMON /WRKTSC/ TSY2,TCY2,TSY3,TCY3
1100      C
1101      C.....+.....+.....+.....+...
1102      C
1103      F(1) = VEL*Y(4)
1104      F(2) = VEL*Y(5)/Y(1)                                !Sngl
1105      C      TSY2 = SIN(Y(2))                                !Sngl
1106      C      TCY2 = COS(Y(2))                                !Dbl
1107      C      TSY2 = DSIN(Y(2))
1108      C      TCY2 = DCOS(Y(2))
1109      C      F(3) = VEL*Y(6)/(Y(1)*TSY2)
1110      C      SQY6 = Y(6)*Y(6)/Y(1)
1111      C      Y5OY1 = Y(5)/Y(1)
1112      C      TAY2 = TSY2/TCY2
1113
1114      C.....+.....+.....+.....+...
1115      C      \/ Use NSSDC routine ALLMAG for magnetic field calculations
1116      C      define MODEL and epoch year (TM)
1117      C      need sine and cosine of phi (Y(3)) for ALLMAG
1118      C      need radial distance from earth center in kilometers (RKM)
1119      C      Remember, ALMAG returns magnetic field in units of Gauss
1120      C.....+.....+.....+...

```

```

1121
1122      MODEL = 14
1123      TM = 1995.0
1124      C   TSY3 = SIN(Y(3))                                !Sngl
1125      C   TCY3 = COS(Y(3))                                !Sngl
1126      C   TSY3 = DSIN(Y(3))                                !Dbl
1127      C   TCY3 = DCOS(Y(3))                                !Dbl
1128      RKM = Y(1)*6371.2
1129
1130      CALL ALLMAG1 (MODEL,TM,RKM,TSY2,TCY2,TSY3,TCY3,BR,BT,BP,B)
1131
1132      c   write (16,1818)  MODEL,TM, Y(1),Y(3),Y(3), RKM,
1133      c   *                               TSY2,TCY2,TSY3,TCY3, BR,BT,BP,B
1134      c1818 format( i5, f8.1, 2x, 3F10.5,2x, f12.3, 2x, 4f10.5,2x, 4f10.5)
1135
1136      F(4) = EOMC*(Y(5)*BP-Y(6)*BT)+VEL*(Y(5)*Y5OY1+SQY6)
1137      F(5) = EOMC*(Y(6)*BR-Y(4)*BP)+VEL*(SQY6/TAY2-Y5OY1*Y(4))
1138      F(6) = EOMC*(Y(4)*BT-Y(5)*BR)-VEL*((Y5OY1*Y(6))/TAY2+Y(4)*Y(6)/
1139      *          Y(1))
1140      RETURN
1141
1142      C.....+.....+.....+.....+.....+.....+.....+...
1143      C   Y(1) is R coordinate           Y(2) is THETA coordinate
1144      C   Y(3) is PHI coordinate        Y(4) is V(R)
1145      C   Y(5) is V(THETA)            Y(6) is V(PHI)
1146      C   F(1) is R dot              F(2) is THETA dot
1147      C   F(3) is PHI dot            F(4) is R dot dot
1148      C   F(5) is THETA dot dot     F(6) is PHI dot dot
1149      C   BR  is B(R)                BT  is B(THETA)
1150      C   BP  is B(PHI)              B   is magnitude of magnetic field
1151
1152      C.....+.....+.....+.....+.....+.....+...
1153      END
1154      C   Program: allmag_sub.f      Version: 1.1      Last Updated: 12/30/97 13:24:51
1155      C   This subroutine contains all the geomagnetic field coefficients.
1156      C   Written in Digital UNIX FORTRAN (12/01/1997)
1157      !*****SUBROUTINE ALLMAG1 !*****!
1158      !
1159      !*****SUBROUTINE ALLMAG1 !*****!
1160      SUBROUTINE ALLMAG1 (MODEL,TM,RKM,ST,CT,SPH,CPH,BR,BT,BP,B)      ALMGL001
1161
1162      C   NOTE ADDITION OF NEXT STATEMENT
1163      C   **** GEOCENTRIC VERSION OF GEOMAGNETIC FIELD ROUTINE      ALMGL002
1164      C   **** LONG DECK, THROUGH NMAX=13, FIXED INDICES WITHOUT DO LOOPS      ALMGL004
1165      C   **** EXECUTION TIME PER CALL FACTOR OF THREE LESS THAN SHORT DECK      ALMGL005
1166      C   **** PROGRAM DESIGNED AND TESTED BY E G STASSINOPoulos AND G D MEAD,      ALMGL006
1167      C   **** CODE 641, NASA GODDARD SPACE FLT CTR, GREENBELT, MD 20771      ALMGL007
1168      C   **** INPUT MODEL      CHOICE OF 14 MODELS - SEE BELOW      ALMGL008
1169      C   **** RKM      GEOCENTRIC DISTANCE IN KILOMETERS      ALMGL009
1170      C   **** TM      TIME IN YEARS FOR DESIRED FIELD      ALMGL010
1171      C   **** ST,CT      SIN + COS OF GEOCENTRIC COLATITUDE      ALMGL011
1172      C   **** SPH,CPH      SIN + COS OF EAST LONGITUDE      ALMGL012
1173      C   **** OUTPUT BR,BT,BP      GEOCENTRIC FIELD COMPONENTS IN GAUSS      ALMGL013
1174      C   **** B      FIELD MAGNITUDE IN GAUSS      ALMGL014
1175
1176      C   IMPLICIT DOUBLE PRECISION(A-H,O-Z)
1177
1178      C   COMMON /TRAJAC/ CONSTEM,T,FILENAM
1179      C   COMMON /DIPOLE/ WLONG,COLAT,EM
1180
1181      DIMENSION T0(14),NMX(14),ISUM(14,3),G(13,13)      ALMGL024
1182
1183      DATA T0 /4*1960.,2*1965.,1970.,1980.,3*1975.,1985.,1990.,1995./,      ALMGL025
1184      $ NMX /10,11,12,11,9,9,13,11,9,13,13,13,13,13/
1185
1186      INTEGER LSUM(14,3)/-1646106,-1795169,-1865298,-1777057,-158472,      ALMGL026
1187      A-156856,-2191704,-1996220,-168051,-2252599,-2445733,-2369772,      ALMGL027
1188      B-2409473,-246795,-62661,-96778,-181519,-83555,-9569,-9599,-8593,
1189      C-5412,-7351,-11947,-10278,-6777,-8938,-12380,1,-10618,7*1,-2698,
1190      D1,1,1,1/

```

```

1191
1192     INTEGER*4 G1(13,13),GT1(13,13),GTT1(13,13),G2(13,13),GT2(13,13), ALMGL029
1193     1 GTT2(13,13),G3(13,13),GT3(13,13),GTT3(13,13),G4(13,13), ALMGL030
1194     2 GT4(13,13),GTT4(13,13),G5(13,13),GT5(13,13),GTT5(13,13), ALMGL031
1195     3 G6(13,13),GT6(13,13),GTT6(13,13),G7(13,13),GT7(13,13),GTT7(13,13) ALMGL032
1196     4 ,G8(13,13),GT8(13,13),GTT8(13,13),
1197     5 G9(13,13),GT9(13,13),GTT9(13,13),
1198     6 G10(13,13),GT10(13,13),GTT10(13,13),
1199     7 G11(13,13),GT11(13,13),GTT11(13,13),
1200     8 G12(13,13),GT12(13,13),GTT12(13,13),
1201     8 G13(13,13),GT13(13,13),GTT13(13,13),
1202     8 G14(13,13),GT14(13,13),GTT14(13,13)
1203     9 ,LG(13,13,14),LGT(13,13,14),LGTT(13,13,14) ALMGL033
1204
1205     REAL*4 GG(13,13,14),GGT(13,13,14),GGTT(13,13,14),SHMIT(13,13) ALMGL034
1206     EQUIVALENCE (G1(1),GG(1),LG(1)), (GT1(1),GGT(1),LGT(1)), ALMGL035
1207     A (GTT1(1),GGTT(1),LGTT(1)),
1208     B (G2(1),LG(1,1,2)), (GT2(1),LGT(1,1,2)), (GTT2(1),LGTT(1,1,2)), ALMGL037
1209     C (G3(1),LG(1,1,3)), (GT3(1),LGT(1,1,3)), (GTT3(1),LGTT(1,1,3)), ALMGL038
1210     D (G4(1),LG(1,1,4)), (GT4(1),LGT(1,1,4)), (GTT4(1),LGTT(1,1,4)), ALMGL039
1211     E (G5(1),LG(1,1,5)), (GT5(1),LGT(1,1,5)), (GTT5(1),LGTT(1,1,5)), ALMGL040
1212     F (G6(1),LG(1,1,6)), (GT6(1),LGT(1,1,6)), (GTT6(1),LGTT(1,1,6)), ALMGL041
1213     G (G7(1),LG(1,1,7)), (GT7(1),LGT(1,1,7)), (GTT7(1),LGTT(1,1,7)), ALMGL042
1214     H (G8(1),LG(1,1,8)), (GT8(1),LGT(1,1,8)), (GTT8(1),LGTT(1,1,8)),
1215     I (G9(1),LG(1,1,9)), (GT9(1),LGT(1,1,9)), (GTT9(1),LGTT(1,1,9)),
1216     J (G10(1),LG(1,1,10)), (GT10(1),LGT(1,1,10)), (GTT10(1),LGTT(1,1,10))
1217     K,(G11(1),LG(1,1,11)),(GT11(1),LGT(1,1,11)),(GTT11(1),LGTT(1,1,11))
1218     L,(G12(1),LG(1,1,12)),(GT12(1),LGT(1,1,12)),(GTT12(1),LGTT(1,1,12))
1219     M,(G13(1),LG(1,1,13)),(GT13(1),LGT(1,1,13)),(GTT13(1),LGTT(1,1,13))
1220     N,(G14(1),LG(1,1,14)),(GT14(1),LGT(1,1,14)),(GTT14(1),LGTT(1,1,14)) ALMGL043
1221
1222     C ***** THE FOLLOWING DATA CARDS CONTAIN THE FIELD COEFFICIENTS ALMGL044
1223     C ***** FOR THE FOLLOWING SEVEN MODELS ALMGL045
1224     C ***** G1,GT1 HENDRICKS + CAIN 99-TERM GSFC 9/65 EPOCH 1960.ALMLG046
1225     C ***** G2,GT2,GTT2 CAIN ET. AL. 120-TERM GSFC 12/66 EPOCH 1960.ALMLG047
1226     C ***** G3,GT3 CAIN + LANGE 143-TERM POGO 10/68 EPOCH 1960.ALMLG048
1227     C ***** G4,GT4 CAIN + SWEENEY 120-TERM POGO 8/69 EPOCH 1960.ALMLG049
1228     C ***** G5,GT5 IGRF 1965.0 80-TERM 10/68 EPOCH 1965.ALMLG050
1229     C ***** G6,GT6 LEATON MALIN + EVANS 1965 80-TERM EPOCH 1965.ALMLG051
1230     C ***** FOR MODEL 6 (LME 1965) SET RKM = 6371.2 + ALTITUDE ALMGL052
1231     C ***** G7,GT7 HURWITZ US COAST + GEODETIC S. 168-TERM EPOCH 1970.ALMLG052
1232     C ***** 8 IGRF 1980 EPOCH980.
1233     C ***** 9 IGRF 1975 80-TERM80-TERM EPOCH 1975.
1234     C ***** 10 BARRACLOUGH 168-TERM "
1235     C ***** 11 AWC "
1236     C ***** 12 IGRF 1985 "
1237     C ***** 13 IGRF 1990 "
1238     C ***** 14 IGRF 1995 "
1239     C HENDRICKS / CAIN ET AL MODEL * 99-TERM GSFC 9/65 EPOCH 1960.0
1240
1241     DATA G1 / 10, -304249,-15361,13009,9576,-2277,498,709,48,99,3*0, ALMGL053
1242     A 57748,-21616,30002,-19870,8028,3595,607,-572,67,29,3*0,-19498, ALMGL054
1243     B 2043,15853,12904,5026,2313,45,56,-88,74,3*0,-4310,2308,-1300,8712ALMGL055
1244     C ,-3940,-312,-2417,75,-138,-156,3*0,1520,-2684,29,-2505,2714, ALMGL056
1245     D -1573,-12,-244,-33,114,3*0,86,1212,-1160,-1104,799,-652,5,-15,71,ALMGL057
1246     E 111,3*0,-119,1028,609,-272,-124,-116,-1091,141,-56,10,3*0,-540, ALMGL058
1247     F -244,-91,22,276,-211,-201,58,117,4*0,69,-122,58,-170,26,236,-25, ALMGL059
1248     G -160,64,16,3*0,-220,156,51,-35,-18,96,121,2,-25,15,42*0 / ALMGL060
1249     DATA GT1 / 100, 2059,-2907,266,-86,255,-70,6*0,-394,602,121,-1003,ALMGL061
1250     H 194,-8,99,6*0,-1369,-1578,-70,163,-117,153,85,6*0,649,293,-924, ALMGL062
1251     I -130,-54,-42,211,6*0,-177,-154,318,-548,-417,-72,157,6*0,304,288,ALMGL063
1252     J -186,125,80,164,-9,6*0,-139,12,153,-73,-6,45,6,84*0/ ALMGL064
1253     DATA GTT1 / 1,168*0/ ALMGL065
1254
1255     C CAIN ET AL 120-TERM GSFC 12/66 EPOCH 1960.0
1256
1257     DATA G2 / 10, -304012,-15401,13071,9493,-2335,492,722,85,104,-29, ALMGL066
1258     A 2*0,57782,-21638,29979,-19889,8035,3557,575,-537,65,58,-9, ALMGL067
1259     B 2*0,-19320,2029,15903,12768,5029,2284,-8,79,-93,75,-22,2*0,-4254,ALMGL068
1260

```

```

1261      C 2278,-1338,8812,-3977,-288,-2383,156,-96,-151,8,2*0,1603,-2743, ALMGL069
1262      D 23,-2466,2665,-1579,-15,-243,-61,121,-28,2*0,51,1178,-1148,-1089,ALMGL070
1263      E 824,-622,-20,-36,55,47,64,2*0,-121,1044,566,-234,-148,-133,-1089,ALMGL071
1264      F 155,-81,2,47,2*0,-537,-274,-81,70,243,-225,-214,36,130,16,-2,2*0,ALMGL072
1265      G 54,-117,42,-153,46,219,-7,-171,74,9,18,2*0,-224,138,63,-30,-19, ALMGL073
1266      H 90,115,1,-15,2,20,2*0,-1,45,-10,26,-44,-13,-36,40,10,-20,11,28*0/ALMGL074
1267      DATA GT2 / 100, 1403,-2329,-93,145,161,-42,-57,35,-10,-1,2*0,-371,ALMGL075
1268      I 876,-9,-1062,90,60,82,-34,50,-13,-13,2*0,-1431,-1662,-456,231, ALMGL076
1269      J -175,334,82,-144,170,-120,88,2*0,520,253,-698,-589,66,-4,235,-90,ALMGL077
1270      K -11,8,-18,2*0,-219,-14,188,-652,-301,-60,83,3,34,-8,17,2*0,224, ALMGL078
1271      L 159,-261,50,-12,176,1,-60,-7,-39,-2,2*0,5,9,255,-119,33,84,23,-17ALMGL079
1272      M ,43,-36,5,2*0,-96,1,43,75,-33,49,90,-64,-15,47,17,2*0,-50,-21,3, ALMGL080
1273      N -79,5,10,-36,-43,-42,37,16,2*0,66,54,3,35,-3,-1,45,-5,75,-46,31, ALMGL081
1274      O 2*0,-61,-64,2,5,-63,-7,7,-3,-2,-45,-23,28*0/ ALMGL082
1275      DATA GTT2 /1000,-62,-154,-123,1,45,-6,-14,6,-5,-3,2*0,-43,114,-18,ALMGL083
1276      P -27,-44,1,15,-6,8,-1,-3,2*0,54,-16,-253,28,17,75,10,-34,39,-27,20,ALMGL084
1277      Q 2*0,95,-7,79,-183,7,8,50,-4,-8,5,-8,2*0,4,56,-35,-47,-97,15,-11, ALMGL085
1278      R -6,15,-7,7,2*0,-46,7,-7,1,-24,56,26,-27,-2,-6,1,2*0,20,-11,15, ALMGL086
1279      S -29,29,-10,23,-1,5,-9,1,2*0,-14,16,14,5,-8,16,11,-4,-8,6,1,2*0, ALMGL088
1280      T -15,-12,5,-11,0,-3,-9,-3,-7,5,5,2*0,22,7,-2,9,6,-1,9,-4,19,-9,4, ALMGL089
1281      U 2*0,-12,-14,1,1,-11,-1,1,-1,1,-6,-2,28*0/
1282
1283      C   CAIN / LANGE * 143-TERM POGO 10/68 EPOCH 1960.0
1284      DATA G3 / 10, -304650,-15414,13258,9591,-2343,491,759,74,110,-26, ALMGL090
1285      A 23,0,57910,-21633,29763,-19837,8196,3577,545,-524,60,66,-20,-18, ALMGL091
1286      B 0,-19772,1566,16075,13169,4864,2339,48,80,-81,18,10,-21,0,-4453, ALMGL092
1287      C 2334,-949,8420,-3724,-210,-2491,100,-92,-125,-55,55,0,1354,-2667,ALMGL093
1288      D 207,-2415,2562,-1471,17,-367,-8,158,-7,-15,0,169,1133,-1287,-1151ALMGL094
1289      E ,1303,-452,-37,-83,91,17,75,24,0,-96,1064,568,-272,-149,-43,-916,ALMGL095
1290      F 66,-114,26,78,-35,0,-579,-250,-8,63,95,-117,-376,-227,79,87,17, ALMGL096
1291      G -13,0,101,-130,115,-164,55,223,-49,-262,351,51,-53,25,0,-204,144,ALMGL097
1292      H 6,-15,14,34,148,24,-9,-24,13,-12,0,11,9,-3,75,-23,14,-5,43,80, ALMGL098
1293      I -137,-27,127,0,-8,44,-1,-39,-6,18,-32,8,-59,-17,105,50,14*0/ ALMGL099
1294      DATA GT3 / 100,2542,-2390,-559,-62,272,-61,-89,61,-24,-1,3,0,-466,ALMGL100
1295      J 988,350,-1152,-251,48,106,-21,-12,30,-9,11,0,-707,-1070,-214,-441ALMGL101
1296      K ,-122,317,62,-108,87,4,12,5,0,848,68,-1489,287,-296,-246,396,70, ALMGL102
1297      L -33,4,19,-30,0,345,-39,-87,-652,86,-89,-94,107,-14,-40,-20,1,0,5,ALMGL103
1298      M 300,32,311,-635,-315,149,96,-85,-28,-2,-34,0,-26,-48,258,-80,50, ALMGL104
1299      N 82,-167,101,99,-57,-43,48,0,-87,-46,-102,25,188,-243,232,523,81, ALMGL105
1300      O -132,-33,52,0,-15,-10,-122,-26,15,-37,29,91,-498,-14,103,-19,0, ALMGL106
1301      P -38,16,67,-14,-83,130,-33,-38,99,50,22,-3,0,21,5,54,-26,-30,-3, ALMGL107
1302      Q -39,-2,-104,79,46,-165,0,35,-26,-17,17,18,-50,23,-34,37,22,-155, ALMGL108
1303      R -40,14*0/ ALMGL109
1304      DATA GTT3 /1,168*0/
1305
1306      C   CAIN / SWEENEY * 120-TERM POGO 8/69 EPOCH 1960.0
1307      DATA G4 / 10,-304708,-15425,13334,9647,-2375,448,793,99,96,-17, ALMGL111
1308      A 2*0,57571,-21702,29893,-19826,8108,3566,594,-516,32,93,-22,2*0, ALMGL112
1309      B -19793,2661,15559,12922,5068,2498,-37,-3,-56,31,13,2*0,-4249, ALMGL113
1310      C 2417,-1740,8336,-3978,-143,-2324,89,-165,-120,16,2*0,1344,-3037, ALMGL114
1311      D 194,-2764,2247,-1497,96,-335,-33,153,-22,2*0,51,1080,-1073,-1083,ALMGL115
1312      E 1171,-757,20,-33,50,7,94,2*0,-76,1181,583,-181,-270,1,-831,100, ALMGL116
1313      F -120,8,87,2*0,-544,-212,-87,55,151,-236,-278,39,102,4,3,2*0,98, ALMGL117
1314      G -162,99,-189,106,206,-2,-207,187,62,-24,2*0,-254,128,31,-25,-21, ALMGL118
1315      H 73,127,47,7,-38,-1,2*0,29,35,-7,66,-50,10,-28,21,42,-88,53,28*0/ ALMGL119
1316      DATA GT4 / 100,2682,-2366,-724,-157,359,12,-160,19,17,-3,2*0,225, ALMGL120
1317      I 1003,150,-1142,-118,58,38,-26,27,-8,-8,2*0,-684,-2832,792,84, ALMGL121
1318      J -536,-27,235,72,33,-46,17,2*0,449,-96,177,327,102,-326,128,86,83,ALMGL122
1319      K -9,-87,2*0,369,564,-109,-205,834,-108,-277,84,42,-37,-12,2*0,234,ALMGL123
1320      L 401,-424,63,-503,504,8,-57,0,-3,-33,2*0,-65,-238,249,-170,234, ALMGL124
1321      M -259,-130,101,49,-48,-33,2*0,-168,-114,58,123,94,40,60,-140,73, ALMGL125
1322      N 54,-21,2*0,1,39,-106,-9,-49,56,-67,-8,-148,-13,27,2*0,48,42,17, ALMGL126
1323      O -41,-22,21,1,-113,16,33,49,2*0,-14,-37,51,-2,4,-19,7,40,-53,31, ALMGL127
1324      P -75,28*0/ ALMGL128
1325      DATA GTT4 /1,168*0/
1326
1327      C   IGRF 1965.0 * 80-TERM 10/68 EPOCH 1965.0
1328
1329
1330

```

```

1331      DATA G5 / 1, -30339, -1654, 1297, 958, -223, 47, 71, 10, 4*0, 5758, -2123, ALMGL130
1332      A 2994, -2036, 805, 357, 60, -54, 9, 4*0, -2006, 130, 1567, 1289, 492, 246, 4, 0, ALMGL131
1333      B -3, 4*0, -403, 242, -176, 843, -392, -26, -229, 12, -12, 4*0, 149, -280, 8, -265ALMGL132
1334      C , 256, -161, 3, -25, -4, 4*0, 16, 125, -123, -107, 77, -51, -4, -9, 7, 4*0, -14, ALMGL133
1335      D 106, 68, -32, -10, -13, -112, 13, -5, 4*0, -57, -27, -8, 9, 23, -19, -17, -2, 12, ALMGL134
1336      E 4*0, 3, -13, 5, -17, 4, 22, -3, -16, 6, 56*0/
1337      DATA GT5 / 10, 153, -244, 2, -7, 19, -1, -5, 1, 4*0, -23, 87, 3, -108, 2, 11, -3, ALMGL136
1338      F -3, 4, 4*0, -118, -167, -16, 7, -30, 29, 11, -7, 6, 4*0, 42, 7, -77, -38, -1, 6, 19, ALMGL137
1339      G -5, 5*0, -1, 16, 29, -42, -21, 0, -4, 3, 5*0, 23, 17, -24, 8, -3, 13, -4, 0, -1, 4*0, ALMGL138
1340      H -9, 4, 20, -11, 1, 9, -2, -2, 3, 4*0, -11, 3, 4, 2, 4, 2, 3, -6, -3, 4*0, 1, -2, -3, -2, ALMGL139
1341      I -3, -4, -3, -3, -5, 56*0/                               ALMGL140
1342      DATA GTT5 /1, 168*0/                               ALMGL141
1343
1344      C LEATON, MALIN + EVANS 1965 * 80-TERM EPOCH 1965.0
1345      DATA G6 / 1, -30375, -1648, 1164, 930, -179, 42, 77, 11, 4*0, 5769, -2087, ALMGL142
1346      A 2954, -2033, 811, 357, 55, -56, 23, 4*0, -1995, 116, 1579, 1299, 490, 248, 12, ALMGL143
1347      B 8, -6, 4*0, -389, 230, -141, 880, -402, -20, -239, 5, -17, 4*0, 142, -276, 5, ALMGL144
1348      C -264, 262, -171, 16, -35, 5, 4*0, 30, 135, -123, -100, 84, -64, 8, -16, 20, 4*0, ALMGL145
1349      D -18, 101, 60, -32, -27, -12, -110, 9, -1, 4*0, -47, -35, -9, 2, 27, -17, -24, 2, ALMGL146
1350      E 12, 4*0, 5, -7, 3, -20, 8, 26, 10, -12, 7, 56*0/
1351      DATA GT6 / 10, 155, -266, 0, 6, 8, 7*0, 6, 83, -13, -95, 10, 4, -5, 6*0, -114, ALMGL148
1352      F -182, 13, -19, -22, 16, 18, 6*0, 32, 16, -85, -6, 2, -3, 14, 6*0, 30, -7, 27, -27, ALMGL149
1353      G -30, -11, 6, 6*0, 19, 23, -18, 14, 5, 17, 2, 6*0, -22, 2, 9, -21, -1, -2, -22, 84*0/ALMGL150
1354      DATA GTT6 /1, 168*0/                               ALMGL151
1355
1356      C HURWITZ (U S COAST / GEODETIC SURVEY) * 168-TERM EPOCH 1970.0
1357
1358      DATA G7/10, -302059, -17917, 12899, 9475, -2145, 460, 734, 121, 107, -39, 16, ALMGL152
1359      A -4, 57446, -20664, 29971, -20708, 8009, 3595, 651, -546, 77, 57, -26, -31, 30, ALMGL153
1360      B -20582, 430, 16086, 12760, 4579, 2490, 95, 46, -32, 23, 7, -36, 5, -3699, 2456, ALMGL154
1361      C -1880, 8334, -3960, -290, -2188, 175, -124, -110, -19, 37, -3, 1617, -2758, ALMGL155
1362      D 185, -2788, 2436, -1669, 20, -210, -44, 131, -15, -3, -13, 157, 1420, -1310, ALMGL156
1363      E -911, 808, -582, -22, -32, 45, 33, 74, -6, 4, -171, 1146, 625, -323, -78, 38, ALMGL157
1364      F -1125, 143, 34, 2, 46, -8, -14, -666, -265, -34, 81, 209, -240, -186, 41, 125, ALMGL158
1365      G 15, 6, 1, -12, 121, -160, 22, -176, 46, 189, -46, -187, 94, 9, -8, 2, -12, -174, ALMGL159
1366      H 163, 14, -27, -32, 80, 137, -4, -14, -4, 22, -24, -1, 27, 19, 0, 35, -45, 22, -31, ALMGL160
1367      I 56, -1, -63, 14, 4, 10, -2, 26, -26, -9, 21, -1, 18, -14, -28, -17, -14, 6, -4, -3, ALMGL161
1368      J 4, 9, -1, -10, 26, -32, 13, -6, -19, 7, 19, 12/
1369      DATA GT7/10, 231, -244, -19, -7, 12, -7, 0, 3, 4*0, -46, 112, -1, -90, -6, 7, 6, ALMGL163
1370      K -3, 3, 4*0, -104, -166, 40, -20, -36, 12, 14, 3, 4, 4*0, 72, 21, -52, -54, -11, 0, ALMGL164
1371      L 17, 6, 1, 4*0, 0, 22, -5, 14, -24, -23, -15, 6, 3, -1, 4*0, 1, 25, -14, 9, 1, 11, -3, 2, ALMGL165
1372      M -3, 4*0, -5, 11, 2, -3, 7, 22, -5, 1, 9, 4*0, -17, -3, 7, 1, -2, -3, -2, -1, -2, 4*0, ALMGL166
1373      N 2, -6, -3, -4, 1, -2, -2, -1, 6, 56*0/                               ALMGL167
1374      DATA GTT7 /1, 168*0/                               ALMGL168
1375
1376      C LANGEL FIELD COEFFICIENTS - 120 TERM POGO 8/71 EPOCH 1960
1377
1378      C THIS WILL BE MODEL 13 IF NEEDED
1379
1380      C DATA G13/10, -304609, -15437, 13085, 9598, -2233, 468, 725, 77, 122, -25,
1381      C A 2*0, 58089, -21750, 29974, -19844, 8125, 3588, 583, -511, 45, 64, -22, 2*0,
1382      C B -19882, 2124, 15676, 13110, 5060, 2408, -15, 46, -59, 30, 33, 2*0, -4408,
1383      C C 2776, -1449, 8684, -3826, -236, -2420, 127, -119, -121, -26, 2*0, 1308,
1384      C D -2806, 32, -2678, 2740, -1513, -18, -352, -18, 151, -20, 2*0, 124, 1156,
1385      C E -1128, -1205, 933, -488, -32, -74, 70, 15, 90, 2*0, -67, 1085, 681, -210, -250,
1386      C F -225, -719, 147, -185, 3, 100, 2*0, -547, -259, -74, 133, 212, -188, -320, 17,
1387      C G 182, 10, -15, 2*0, 107, -135, 60, -182, 124, 234, 2, -231, 209, 68, -19, 2*0,
1388      C H -200, 155, 13, -52, 5, 94, 152, -4, -78, 28, 40, 2*0, 30, 19, 11, 61, -56, 6, -50,
1389      C I 50, 6, -35, -11, 28*0/
1390
1391      C DATA GT13 / 10, 245, -234, -32, -9, 12, 0, -4, 4, -1, 0, 2*0, -63, 104, 3, -111,
1392      C J -12, 1, 4, -3, 1, 2, 0, 2*0, -50, -203, 39, -16, -44, 10, 18, 0, 4, -4, -1, 2*0, 71,
1393      C K 15, -17, -33, -12, -14, 27, 2, 1, 1, -2, 2*0, 38, 21, 9, -29, 13, -9, -11, 11, 1, -4,
1394      C L 0, 2*0, 12, 25, -36, 19, -15, 16, 7, 0, -2, -1, -2, 2*0, -6, -9, 12, -13, 20, 8, -27,
1395      C M 2, 14, -3, -5, 2*0, -16, -2, 4, 3, 1, -1, 13, -9, -2, 3, 0, 2*0, -2, 0, -5, -1, -7, 0,
1396      C N -6, 3, -15, -1, 1, 2*0, -1, 1, 3, -1, -5, -1, -4, -3, 12, -5, -1, 2*0, -1, 0, 3, 0,
1397      C O 1, -1, 3, 0, 0, -3, 1, 28*0/
1398
1399      C IGRF 1980 FIELD COEFFICIENTS (MODEL = 8)
1400

```

```

1401      DATA G8/10,-299880,-19970,12790,9380,-2190,490,700,200,60,-30,0,0,
1402      A 56060,-19570,30280,-21810,7830,3570,650,-590,70,110,-40,0,0,
1403      B -21290,-1990,16620,12510,3980,2610,420,20,10,20,20,0,0,
1404      C -3350,2710,-2520,8330,-4190,-740,-1920,200,-110,-120,-50,0,0,
1405      D 2120,-2570,530,-2980,1990,-1620,40,-130,-70,90,-20,0,0,
1406      E 460,1490,-1500,-780,920,-480,140,10,40,-30,50,0,0,
1407      F -150,930,710,-430,-20,170,-1080,110,30,-10,30,0,0,
1408      G -830,-280,-50,160,180,-230,-100,-20,70,70,10,0,0,
1409      H 70,-180,40,-220,90,160,-130,-150,-10,10,20,0,0,
1410      I -210,160,90,-50,-70,90,100,-60,20,-50,30,0,0,
1411      J 10,10,20,50,-40,-10,-20,40,-10,-60,29*0/
1412
1413      DATA GT8 /10,224,-183,0,-14,15,4,-10,8,0,0,2*0,
1414      A -159,113,32,-65,-14,4,0,-8,-2,0,0,2*0,
1415      B -127,-252,70,-7,-82,-8,34,4,-3,0,0,2*0,
1416      C 2,27,-79,10,-18,-33,8,5,3,0,0,2*0,
1417      D 46,16,29,4,-50,2,8,16,-8,0,0,2*0,
1418      E 18,-4,0,13,21,14,3,1,-2,0,0,2*0,
1419      F -5,-14,0,-16,5,0,-1,1,7,0,0,2*0,
1420      G -4,4,2,14,-5,-1,11,0,-3,0,0,2*0,
1421      H -1,-7,0,-8,2,2,-11,8,12,0,0,2*0,
1422      I 52*0/
1423
1424      DATA GTT8 / 1,168*0/
1425
1426      C IGRF 1975 FIELD COEFFICIENTS (MODEL = 9)
1427
1428      DATA G9 /1,-30186,-1898,1299,951,-204,46,66,11,4*0,5735,-2036,
1429      A2997,-2144,807,368,57,-57,13,4*0,-2124,-37,1551,1296,462,275,15,
1430      B-7,3,4*0,-361,249,-253,805,-393,-20,-210,7,-12,4*0,148,-264,37,
1431      C-307,235,-161,-1,-22,-4,4*0,39,142,-147,-99,74,-38,-8,-9,6,4*0,
1432      D-23,102,88,-43,-9,-4,-114,11,-2,4*0,-68,-24,-4,11,27,-17,-14,-8,
1433      E9,4*0,4,-15,2,-19,1,18,-6,-19,1,56*0/
1434
1435      DATA GT9 /10,256,-249,-38,-2,3,2,0,2,4*0,-102,100,7,-104,-20,-7,
1436      F5,0,3,4*0,-30,-189,43,-41,-30,11,20,6*0,69,25,-50,-42,-21,-16,28,
1437      G6,2,4*0,50,8,17,-10,-31,-5,0,9,-4,4*0,12,23,-20,13,11,10,9,3,-3,
1438      H4*0,-5,-1,-2,-13,7,17,-1,3,6,4*0,-14,-1,3,3,-7,1,8,-5,-3,4*0,-2,
1439      I-4,-2,-3,4,-3,-6,3,-1,56*0/
1440
1441      DATA GTT9 /1,168*0/
1442
1443      C BARRACLOUGH FIELD COEFFICIENTS (MODEL = 10)
1444
1445      DATA G10/10,-301036,-19067,12782,9469,-2206,441,715,110,93,-50,28,
1446      A-5,56826,-20165,30099,-21420,7925,3514,699,-533,51,100,-33,-19,8,
1447      B-20647,-581,16330,12547,4438,2623,277,23,-26,16,24,-45,-13,-3298,
1448      C2659,-2270,8310,-4039,-638,-1943,134,-126,-114,-60,29,1,1934,-2658
1449      D,530,-2852,2125,-1575,-9,-64,-138,106,-14,-12,-6,245,1484,-1613,
1450      E-834,923,-402,38,32,-1,6,66,13,0,-112,1004,776,-403,-79,156,-1087,
1451      F170,-24,-2,46,-8,-18,-766,-247,-45,70,245,-218,-129,-59,123,6,12,
1452      G19,-16,49,-139,50,-180,57,145,-111,-167,49,5,-18,34,-7,-196,157,49
1453      H,-31,-42,97,122,-2,3,5,34,-16,-9,13,20,26,28,-36,-2,3,32,30,-34,
1454      I-10,17,4,5,7,-9,-15,3,6,-21,9,-25,-7,3,25,0,1,-5,4,0,8,-1,-2,-5,3,
1455      J-20,14,-2,11/
1456
1457      DATA GT10/10,268,-250,-38,-9,2,6,-4,4,4*0,-101,100,3,-105,-22,-10,
1458      K9,-2,3,4*0,-28,-189,55,-47,-40,13,23,-5,0,4*0,72,28,-64,-47,-21,
1459      L-21,35,3,4,4*0,54,7,26,-7,-46,-6,0,8,-2,4*0,9,26,-27,13,11,13,8,6,
1460      M-4,4*0,-3,-2,2,-16,4,20,-4,5,6,4*0,-12,-2,0,3,-6,0,12,-8,-3,4*0,
1461      N-2,-3,-3,5,-5,-6,5,0,56*0/
1462
1463      DATA GTT10/100,70,-20,-28,0,-13,7,6*0,-49,0,0,0,-17,-14,4,6*0,68,
1464      O0,16,-32,0,-14,12,6*0,13,0,0,-14,-10,11,6*0,30,0,0,16,-17,0,0,
1465      P6*0,-14,10,0,0,10,0,9,6*0,0,0,-10,88*0/
1466
1467      C AWC FIELD COEFFICIENTS (MODEL = 11)
1468
1469      DATA G11/10,-300557,-19320,12671,9538,-2142,417,743,124,125,-55,
1470

```

```

1471      A44,-7,56705,-20170,30013,-21272,7861,357,642,-497,84,52,-20,-31,
1472      B0,-20444,-692,16197,12594,4378,2561,183,54,-37,15,20,-27,16,
1473      C-3435,2632,-2085,8180,-4128,-428,-1994,248,-117,-74,-27,50,4,1967,
1474      D-2570,201,-2875,2323,-1667,32,-118,-85,119,-40,-3,-15,314,1507,
1475      E-1374,-819,862,-589,106,-37,59,23,91,-13,-12,-199,1056,608,-392,
1476      F18,146,-1085,150,29,-4,42,-10,-4,-730,-279,-52,82,139,-210,-93,
1477      G3,71,29,-22,10,-36,69,-154,48,-170,110,155,-96,-191,31,16,-8,9,6,
1478      H-172,172,42,-1,-49,91,122,-29,8,-8,50,-21,8,2,6,-5,43,-40,27,-20,
1479      I43,1,-45,21,8,-3,10,24,-21,24,32,17,-7,-30,-21,2,-1,-5,13,13,-18,
1480      J9,1,-18,14,-11,1,19,-33,6,15,7/
1481
1482      DATA GT11 /0,244,-249,-37,5,3,-3,4,5*0,-103,99,12,-104,-18,-3,1,
1483      K2,4,4*0,-31,-190,31,-34,-37,10,17,6,1,4*0,67,21,-35,-37,-21,-12,
1484      L21,9,-1,4*0,47,10,9,-14,-16,-4,-1,9,-5,4*0,15,20,-13,13,10,6,9,0,
1485      M-3,4*0,-7,0,-6,-10,11,15,2,1,5,4*0,-15,0,5,2,-8,2,4,-2,-4,4*0,-2,
1486      N-4,0,-3,4,-2,-7,0,-2,56*0/
1487
1488      DATA GTT11 /1,168*0/
1489
1490      C   IGRF 1985 FIELD COEFFICIENTS (MODEL = 12)
1491
1492      DATA G12 / 10,-298770,-20730,13000,9370,-2150,520,750,210,50,-40,
1493      $2*0,54970,-19030,30450,-22080,7800,3560,650,-610,60,100,-40,2*0,
1494      $-21910,-3090,16910,12440,3630,2530,500,2,0,10,20,2*0,
1495      $-3120,2840,-2960,8350,-4260,-940,-1860,240,-110,-120,-50,2*0,
1496      $2330,-2500,680,-2980,1690,-1610,40,-60,-90,90,-20,2*0,
1497      $470,1480,-1550,-750,950,-480,170,40,20,-30,50,2*0,
1498      $-160,900,690,-500,-40,200,-1020,90,40,-10,30,2*0,
1499      $-820,-260,-10,230,170,-210,-60,0,40,70,10,2*0,
1500      $70,-210,50,-250,110,120,-160,-100,-60,20,20,2*0,
1501      $-210,160,90,-50,-60,90,100,-50,20,-50,30,2*0,
1502      $10,0,30,60,-40,0,-10,40,0,-60,0,28*0/
1503
1504      DATA GT12 / 10,232,-137,51,1,13,14,2,7,4*0,
1505      $-245,100,34,-46,-6,1,-3,-6,0,4*0,
1506      $-115,-202,70,-6,-78,-15,17,-5,3,4*0,
1507      $53,23,-108,1,-14,-32,6,8,4,4*0,
1508      $38,22,25,9,-68,1,0,10,-3,4*0,
1509      $1,-2,-1,6,0,-1,9,4,-3,4*0,
1510      $-4,-11,-8,-23,-5,-1,12,-5,1,4*0,
1511      $2,10,11,19,3,2,9,-1,-5,4*0,
1512      $1,-10,1,-8,2,-8,-1,13,-8,4*0,52*0/
1513
1514      DATA GTT12 / 1,168*0/
1515
1516      C   IGRF 1990 COEFFICIENTS (MODEL = 13)
1517
1518      DATA G13/10,-297754,-21358,13146,9389,-2110,607,766,224,44,-36,
1519      *0,0,
1520      A54109,-18510,30582,-22402,7823,3525,639,-642,51,99,-39,0,0,
1521      B-22777,-3800,16932,12456,3239,2438,604,37,-9,8,24,0,0,
1522      C-2865,2933,-3485,8065,-4227,-1108,-1775,275,-108,-120,-53,0,0,
1523      D2481,-2395,870,-2994,1417,-1656,20,9,-124,93,-24,0,0,
1524      E472,1535,-1544,-692,977,-370,167,57,38,-39,44,0,0,
1525      F-158,827,683,-525,18,269,-963,98,38,-14,30,0,0,
1526      G-811,-273,6,204,164,-226,-50,-5,26,73,12,0,0,
1527      H97,-199,71,-221,119,110,-160,-107,-60,15,22,0,0,
1528      I-208,154,95,-57,-64,86,91,-66,19,-55,29,0,0,
1529      J13,4,31,56,-42,-5,-15,38,-5,-62,29*0/
1530
1531      DATA GT13/10,180,-129,33,5,6,13,6,2,4*0,
1532      A -161,106,24,-67,6,-1,-2,-5,-7,4*0,
1533      B -158,-138,0,0,-70,-16,18,-3,-2,4*0,
1534      C 44,16,-106,-59,5,-31,13,6,1,4*0,
1535      D 26,18,31,-14,-55,0,-2,16,-11,4*0,
1536      E -1,5,4,17,4,23,1,2,0,4*0,
1537      F 2,-13,0,-9,5,12,12,2,0,4*0,
1538      G 6,2,8,-5,-2,0,0,3,-5,4*0,
1539      H 5,-2,3,3,4,-5,-3,6,-6,4*0,52*0/
1540

```

```

1541      DATA GTT13 /1,168*0/
1542
1543 C   IGRF 1995 COEFFICIENTS (MODEL = 14)
1544
1545      DATA G14/1,-29682,-2197,1329,941,-210,66,78,24,4,-3,0,0,
1546      A5318,-1789,3074,-2268,782,352,64,-67,4,9,-4,0,0,
1547      B-2356,-425,1685,1249,291,237,65,1,-1,1,2,0,0,
1548      C-263,302,-406,769,-421,-122,-172,29,-9,-12,-5,0,0,
1549      D262,-232,98,-301,116,-167,2,4,-14,9,-2,0,0,
1550      E44,157,-152,-64,99,-26,17,8,4,-4,4,0,0,
1551      F-16,77,67,-57,4,28,-94,10,5,-2,3,0,0,
1552      G-77,-25,3,22,16,-23,-3,-2,0,7,1,0,0,
1553      H12,-20,7,-21,12,10,-17,-10,-7,0,3,0,0,
1554      I-19,15,11,-7,-7,9,7,-8,1,-6,3,0,0,
1555      J2,1,3,6,-4,0,-2,3,-1,-6,29*0/
1556
1557      DATA GT14/10,176,-132,15,8,8,5,-2,3,4*0,
1558      A -183,130,37,-64,9,1,-4,-8,-2,4*0,
1559      B -150,-88,-8,-2,-69,-15,6,-6,1,4*0,
1560      C 41,22,-121,-81,5,-20,19,6,4,4*0,
1561      D 18,12,27,-10,-46,-1,-2,12,-11,4*0,
1562      E 2,12,3,18,9,23,-2,1,3,4*0,
1563      F 3,-16,-2,-9,10,22,0,2,2,4*0,
1564      G 8,2,6,-4,0,-3,0,-6,-9,4*0,
1565      H 4,-2,2,7,0,-12,-7,-6,-3,4*0,52*0/
1566
1567      DATA GTT14 /1,168*0/
1568
1569 C   DATA SHMIT(1,1)/0.0/,TMOLD/0.0/,MODOLD /0/,RAD/57.29578/ ALMGL169
1570 C   ***** NON-SUBSCRIPTED, FIXED-INDEX VERSION BEGINS HERE (NO DO-LOOPS) ALMGL170
1571 C   ***** BEGIN PROGRAM ALMGL171
1572 C   IF (SHMIT(1,1).EQ.-1.) GO TO 8 ALMGL172
1573 C   ***** INITIALIZE * ONCE ONLY, FIRST TIME SUBROUTINE IS CALLED ALMGL173
1574 C   SHMIT(1,1)=-1. ALMGL174
1575 DO 2 N=2,13 ALMGL175
1576 SHMIT(N,1) = (2*N-3) * SHMIT(N-1,1) / (N-1) ALMGL176
1577 JJ=2 ALMGL177
1578 DO 2 M=2,N ALMGL178
1579 SHMIT(N,M) = SHMIT(N,M-1) * DSQRT(1.0D0*FLOAT((N-M+1)*JJ)/(N+M-2)) ALMGL179
1580 SHMIT(M-1,N)=SHMIT(N,M) ALMGL180
1581 2 JJ = 1 ALMGL181
1582 DO 7 K=1,14 ALMGL182
1583 F1=LG(1,1,K) ALMGL183
1584 F2=LGT(1,1,K) ALMGL184
1585 F3=LGTT(1,1,K) ALMGL185
1586 NMAX=NMX(K) ALMGL186
1587 L = 0 ALMGL187
1588 DO 3 I=1,3 ALMGL188
1589 3 ISUM(K,I) = 0 ALMGL189
1590 DO 4 N=1,NMAX ALMGL190
1591 DO 4 M=1,NMAX ALMGL191
1592 L = L+1 ALMGL192
1593 ISUM(K,1)=ISUM(K,1)+L*LG(N,M,K) ALMGL193
1594 ISUM(K,2)=ISUM(K,2)+L*LGT(N,M,K) ALMGL194
1595 4 ISUM(K,3)=ISUM(K,3)+L*LGTT(N,M,K) ALMGL195
1596 DO 6 I=1,3 ALMGL196
1597 IF (ISUM(K,I).EQ.LSUM(K,I)) GO TO 6 ALMGL197
1598 C   ***** ERROR IN DATA CARDS - NOTE WRITE AND STOP STATEMENTS ALMGL198
1599 PRINT 5, K,I,LSUM(K,I),ISUM(K,I) ALMGL199
1600 5 FORMAT(//29H DATA WRONG IN ALLMAG--MODEL ,I2,3X,2HI=,I1,3X, ALMGL200
1601 A17HPRECALCULATED SUM,I10,3X,17HTHIS MACHINE GETS,I10) ALMGL201
1602 STOP ALMGL202
1603 6 CONTINUE ALMGL203
1604 DO 7 N=1,NMAX ALMGL204
1605 DO 7 M=1,NMAX ALMGL205
1606 GG(N,M,K)=LG(N,M,K)*SHMIT(N,M)/F1 ALMGL206
1607 GGT(N,M,K)=LGT(N,M,K)*SHMIT(N,M)/F2 ALMGL207
1608 7 GGTT(N,M,K)=LGTT(N,M,K)*SHMIT(N,M)/F3 ALMGL208
1609 8 IF ((MODEL.EQ.MODOLD).AND.(TM.EQ.TMOLD)) GO TO 11 ALMGL209
1610 C   ***** NOTE WRITE STATEMENT - NEW MODEL OR NEW TIME ALMGL210

```

```

1611 C      PRINT 9, MODEL, TM          ALMGL211
1612 C      9 FORMAT('0 MODEL USED IS NUMBER ',I2,2X,' FOR TM =',F9.3/) ALMGL212
1613 IF(MODEL.LT.1.OR.MODEL.GT.14) STOP ALMGL213
1614 MODOLD=MODEL ALMGL214
1615 TMOLD=TM ALMGL215
1616 NMAX=NMX(MODEL) ALMGL216
1617 T=TM-T0(MODEL) ALMGL217
1618 DO 10 N=1,NMAX ALMGL218
1619 DO 10 M=1,NMAX ALMGL219
1620 10 G(N,M)=GG(N,M,MODEL)+T*(GGT(N,M,MODEL)+GGTT(N,M,MODEL)*T) ALMGL220
1621 C ***** CALCULATION USUALLY BEGINS
1622 WLONG = -RAD * DATAN (G(1,2) / G(2,2)) ALMGL221
1623 COLAT = RAD * DATAN (SQRT(G(1,2)**2 + G(2,2)**2) / G(2,1)) ALMGL222
1624 EM = DSQRT (G(1,2)**2 + G(2,2)**2 + G(2,1)**2) ALMGL223
1625 CONSTEM = EM / 100000.0 ALMGL224
1626 C      PRINT 19, WLONG, COLAT, EM
1627 C      19 FORMAT(5X,'GEOGRAPHIC COORDINATES OF BOREAL MAGNETIC DIPOLE POLE'/
1628 C      $10X,'WEST LONGITUDE =',F9.3/10X,'GEOC. COLATITUDE =',F9.3/
1629 C      $10X,'EARTH'S MAGNETIC MOMENT =',F8.0,' GAMMA') ALMGL225
1630 11 P21=CT ALMGL226
1631 P22=ST ALMGL227
1632 AR=6371.2/RKM ALMGL228
1633 SP2=SPH ALMGL229
1634 CP2=CPH ALMGL230
1635 DP21=-P22 ALMGL231
1636 DP22=P21 ALMGL232
1637 AOR=AR*AR*AR ALMGL233
1638 C2=G(2,2)*CP2+G(1,2)*SP2 ALMGL234
1639 BR=-(AOR+AOR)*(G(2,1)*P21+C2*P22) ALMGL235
1640 BT=AOR*(G(2,1)*DP21+C2*DP22) ALMGL236
1641 BP=AOR*(G(1,2)*CP2-G(2,2)*SP2)*P22 ALMGL237
1642 IF (NMAX.LE. 2) GO TO 1 N= 3 ALMGL238
1643 C
1644 SP3=(SP2+SP2)*CP2 ALMGL239
1645 CP3=(CP2+SP2)*(CP2-SP2) ALMGL240
1646 P31=P21*P21-0.3333333333 ALMGL241
1647 P32=P21*P22 ALMGL242
1648 P33=P22*P22 ALMGL243
1649 DP31=-P32-P32 ALMGL244
1650 DP32=P21*P21-P33 ALMGL245
1651 DP33=-DP31 ALMGL246
1652 AOR=AOR*AR ALMGL247
1653 C2=G(3,2)*CP2+G(1,3)*SP2 ALMGL248
1654 C3=G(3,3)*CP3+G(2,3)*SP3 ALMGL249
1655 BR=BR-3.0*AOR*(G(3,1)*P31+C2*P32+C3*P33) ALMGL250
1656 BT=BT+AOR*(G(3,1)*DP31+C2*DP32+C3*DP33) ALMGL251
1657 BP=BP-AOR*((G(3,2)*SP2-G(1,3)*CP2)*P32+2.0*(G(3,3)*SP3-G(2,3)*CP3)) ALMGL252
1658 1*P33) ALMGL253
1659 IF (NMAX.LE. 3) GO TO 1 N= 4 ALMGL254
1660 C
1661 SP4=SP2*CP3+CP2*SP3 ALMGL255
1662 CP4=CP2*CP3-SP2*SP3 ALMGL256
1663 P41=P21*P31-0.2666666666666666*P21 ALMGL257
1664 DP41=P21*DP31+DP21*P31-0.2666666666666666*DP21 ALMGL258
1665 P42=P21*P32-0.2000000000000000*P22 ALMGL259
1666 DP42=P21*DP32+DP21*P32-0.2000000000000000*DP22 ALMGL260
1667 P43=P21*P33 ALMGL261
1668 DP43=P21*DP33+DP21*P33 ALMGL262
1669 P44=P22*P33 ALMGL263
1670 DP44=3.0*P43 ALMGL264
1671 AOR=AOR*AR ALMGL265
1672 C2=G(4,2)*CP2+G(1,4)*SP2 ALMGL266
1673 C3=G(4,3)*CP3+G(2,4)*SP3 ALMGL267
1674 C4=G(4,4)*CP4+G(3,4)*SP4 ALMGL268
1675 BR=BR-4.0*AOR*(G(4,1)*P41+C2*P42+C3*P43+C4*P44) ALMGL269
1676 BT=BT+AOR*(G(4,1)*DP41+C2*DP42+C3*DP43+C4*DP44) ALMGL270
1677 BP=BP-AOR*((G(4,2)*SP2-G(1,4)*CP2)*P42+2.0*(G(4,3)*SP3-G(2,4)*CP3)) ALMGL271
1678 1*P43+3.0*(G(4,4)*SP4-G(3,4)*CP4)*P44) ALMGL272
1679 IF (NMAX.LE. 4) GO TO 1 N= 5 ALMGL273
1680 C

```

1681	SP5=(SP3+SP3)*CP3	ALMGL273
1682	CP5=(CP3+SP3)*(CP3-SP3)	ALMGL274
1683	P51=P21*P41-0.25714285*P31	ALMGL275
1684	DP51=P21*DP41+DP21*P41-0.25714285*DP31	ALMGL276
1685	P52=P21*P42-0.22857142*P32	ALMGL277
1686	DP52=P21*DP42+DP21*P42-0.22857142*DP32	ALMGL278
1687	P53=P21*P43-0.14285714*P33	ALMGL279
1688	DP53=P21*DP43+DP21*P43-0.14285714*DP33	ALMGL280
1689	P54=P21*P44	ALMGL281
1690	DP54=P21*DP44+DP21*P44	ALMGL282
1691	P55=P22*P44	ALMGL283
1692	DP55=4.0*P54	ALMGL284
1693	AOR=AOR*AR	ALMGL285
1694	C2=G(5,2)*CP2+G(1,5)*SP2	ALMGL286
1695	C3=G(5,3)*CP3+G(2,5)*SP3	ALMGL287
1696	C4=G(5,4)*CP4+G(3,5)*SP4	ALMGL288
1697	C5=G(5,5)*CP5+G(4,5)*SP5	ALMGL289
1698	BR=BR-5.0*AOR*(G(5,1)*P51+C2*P52+C3*P53+C4*P54+C5*P55)	ALMGL290
1699	BT=BT+AOR*(G(5,1)*DP51+C2*DP52+C3*DP53+C4*DP54+C5*DP55)	ALMGL291
1700	BP=BP-AOR*((G(5,2)*SP2-G(1,5)*CP2)*P52+2.0*(G(5,3)*SP3-G(2,5)*CP3))	ALMGL292
1701	1*P53+3.0*(G(5,4)*SP4-G(3,5)*CP4)*P54+4.0*(G(5,5)*SP5-G(4,5)*CP5)*PALMGL293	ALMGL294
1702	255)	ALMGL295
1703	IF (NMAX.LE. 5) GO TO 1	N= 6 ALMGL296
1704	C	
1705	SP6=SP2*CP5+CP2*SP5	ALMGL297
1706	CP6=CP2*CP5-SP2*SP5	ALMGL298
1707	P61=P21*P51-0.25396825*P41	ALMGL299
1708	DP61=P21*DP51+DP21*P51-0.25396825*DP41	ALMGL300
1709	P62=P21*P52-0.23809523*P42	ALMGL301
1710	DP62=P21*DP52+DP21*P52-0.23809523*DP42	ALMGL302
1711	P63=P21*P53-0.19047619*P43	ALMGL303
1712	DP63=P21*DP53+DP21*P53-0.19047619*DP43	ALMGL304
1713	P64=P21*P54-0.11111111*P44	ALMGL305
1714	DP64=P21*DP54+DP21*P54-0.11111111*DP44	ALMGL306
1715	P65=P21*P55	ALMGL307
1716	DP65=P21*DP55+DP21*P55	ALMGL308
1717	P66=P22*P55	ALMGL309
1718	DP66=5.0*P65	ALMGL310
1719	AOR=AOR*AR	ALMGL311
1720	C2=G(6,2)*CP2+G(1,6)*SP2	ALMGL312
1721	C3=G(6,3)*CP3+G(2,6)*SP3	ALMGL313
1722	C4=G(6,4)*CP4+G(3,6)*SP4	ALMGL314
1723	C5=G(6,5)*CP5+G(4,6)*SP5	ALMGL315
1724	C6=G(6,6)*CP6+G(5,6)*SP6	ALMGL316
1725	BR=BR-6.0*AOR*(G(6,1)*P61+C2*P62+C3*P63+C4*P64+C5*P65+C6*P66)	ALMGL317
1726	BT=BT+AOR*(G(6,1)*DP61+C2*DP62+C3*DP63+C4*DP64+C5*DP65+C6*DP66)	ALMGL318
1727	BP=BP-AOR*((G(6,2)*SP2-G(1,6)*CP2)*P62+2.0*(G(6,3)*SP3-G(2,6)*CP3))	ALMGL319
1728	1*P63+3.0*(G(6,4)*SP4-G(3,6)*CP4)*P64+4.0*(G(6,5)*SP5-G(4,6)*CP5)*PALMGL320	ALMGL321
1729	265+5.0*(G(6,6)*SP6-G(5,6)*CP6)*P66)	ALMGL322
1730	IF (NMAX.LE. 6) GO TO 1	N= 7 ALMGL323
1731	C	
1732	SP7=(SP4+SP4)*CP4	ALMGL324
1733	CP7=(CP4+SP4)*(CP4-SP4)	ALMGL325
1734	P71=P21*P61-0.25252525*P51	ALMGL326
1735	DP71=P21*DP61+DP21*P61-0.25252525*DP51	ALMGL327
1736	P72=P21*P62-0.24242424*P52	ALMGL328
1737	DP72=P21*DP62+DP21*P62-0.24242424*DP52	ALMGL329
1738	P73=P21*P63-0.21212121*P53	ALMGL330
1739	DP73=P21*DP63+DP21*P63-0.21212121*DP53	ALMGL331
1740	P74=P21*P64-0.16161616*P54	ALMGL332
1741	DP74=P21*DP64+DP21*P64-0.16161616*DP54	ALMGL333
1742	P75=P21*P65-0.09090909*P55	ALMGL334
1743	DP75=P21*DP65+DP21*P65-0.09090909*DP55	ALMGL335
1744	P76=P21*P66	ALMGL336
1745	DP76=P21*DP66+DP21*P66	ALMGL337
1746	P77=P22*P66	ALMGL338
1747	DP77=6.0*P76	ALMGL339
1748	AOR=AOR*AR	ALMGL340
1749	C2=G(7,2)*CP2+G(1,7)*SP2	ALMGL341
1750	C3=G(7,3)*CP3+G(2,7)*SP3	ALMGL342

```

1751      C4=G(7,4)*CP4+G(3,7)*SP4          ALMGL343
1752      C5=G(7,5)*CP5+G(4,7)*SP5          ALMGL344
1753      C6=G(7,6)*CP6+G(5,7)*SP6          ALMGL345
1754      C7=G(7,7)*CP7+G(6,7)*SP7          ALMGL346
1755      BR=BR-7.0*AOR*(G(7,1)*P71+C2*P72+C3*P73+C4*P74+C5*P75+C6*P76+C7*P7)ALMGL347
1756      17)                                ALMGL348
1757      BT=BT+AOR*(G(7,1)*DP71+C2*DP72+C3*DP73+C4*DP74+C5*DP75+C6*DP76+C7*DP7)ALMGL349
1758      1DP77)                                ALMGL350
1759      BP=BP-AOR*((G(7,2)*SP2-G(1,7)*CP2)*P72+2.0*(G(7,3)*SP3-G(2,7)*CP3)ALMGL351
1760
1761      1*P73+3.0*(G(7,4)*SP4-G(3,7)*CP4)*P74+4.0*(G(7,5)*SP5-G(4,7)*CP5)*PALMGL352
1762      275+5.0*(G(7,6)*SP6-G(5,7)*CP6)*P76+6.0*(G(7,7)*SP7-G(6,7)*CP7)*P77ALMGL353
1763      3)
1764      IF (NMAX.LE. 7) GO TO 1           N= 8      ALMGL354
1765      C
1766      SP8=SP2*CP7+CP2*SP7              ALMGL355
1767      CP8=CP2*CP7-SP2*SP7              ALMGL356
1768      P81=P21*P71-0.25174825*P61      ALMGL357
1769      DP81=P21*DP71+DP21*P71-0.25174825*DP61   ALMGL358
1770      P82=P21*P72-0.24475524*P62      ALMGL359
1771      DP82=P21*DP72+DP21*P72-0.24475524*DP62   ALMGL360
1772      P83=P21*P73-0.22377622*P63      ALMGL361
1773      DP83=P21*DP73+DP21*P73-0.22377622*DP63   ALMGL362
1774      P84=P21*P74-0.18881118*P64      ALMGL363
1775      DP84=P21*DP74+DP21*P74-0.18881118*DP64   ALMGL364
1776      P85=P21*P75-0.13986013*P65      ALMGL365
1777      DP85=P21*DP75+DP21*P75-0.13986013*DP65   ALMGL366
1778      P86=P21*P76-0.07692307*P66      ALMGL367
1779      DP86=P21*DP76+DP21*P76-0.07692307*DP66   ALMGL368
1780      P87=P21*P77
1781      DP87=P21*DP77+DP21*P77
1782      P88=P22*P77
1783      DP88=7.0*P87
1784      AOR=AOR*AR
1785      C2=G(8,2)*CP2+G(1,8)*SP2          ALMGL369
1786      C3=G(8,3)*CP3+G(2,8)*SP3          ALMGL370
1787      C4=G(8,4)*CP4+G(3,8)*SP4          ALMGL371
1788      C5=G(8,5)*CP5+G(4,8)*SP5          ALMGL372
1789      C6=G(8,6)*CP6+G(5,8)*SP6          ALMGL373
1790      C7=G(8,7)*CP7+G(6,8)*SP7          ALMGL374
1791      C8=G(8,8)*CP8+G(7,8)*SP8          ALMGL375
1792      BR=BR-8.0*AOR*(G(8,1)*P81+C2*P82+C3*P83+C4*P84+C5*P85+C6*P86+C7*P8)ALMGL383
1793      17+C8*P88)
1794      BT=BT+AOR*(G(8,1)*DP81+C2*DP82+C3*DP83+C4*DP84+C5*DP85+C6*DP86+C7*DP87)ALMGL384
1795      1DP87+C8*DP88)
1796      BP=BP-AOR*((G(8,2)*SP2-G(1,8)*CP2)*P82+2.0*(G(8,3)*SP3-G(2,8)*CP3)ALMGL385
1797      1*P83+3.0*(G(8,4)*SP4-G(3,8)*CP4)*P84+4.0*(G(8,5)*SP5-G(4,8)*CP5)*PALMGL386
1798      285+5.0*(G(8,6)*SP6-G(5,8)*CP6)*P86+6.0*(G(8,7)*SP7-G(6,8)*CP7)*P87ALMGL387
1799      3+7.0*(G(8,8)*SP8-G(7,8)*CP8)*P88)
1800      IF (NMAX.LE. 8) GO TO 1           N= 9      ALMGL388
1801      C
1802      SP9=(SP5+SP5)*CP5
1803      CP9=(CP5+SP5)*(CP5-SP5)          ALMGL389
1804      P91=P21*P81-0.25128205*P71      ALMGL390
1805      DP91=P21*DP81+DP21*P81-0.25128205*DP71   ALMGL391
1806      P92=P21*P82-0.24615384*P72      ALMGL392
1807      DP92=P21*DP82+DP21*P82-0.24615384*DP72   ALMGL393
1808      P93=P21*P83-0.23076923*P73      ALMGL394
1809      DP93=P21*DP83+DP21*P83-0.23076923*DP73   ALMGL395
1810      P94=P21*P84-0.20512820*P74      ALMGL396
1811      DP94=P21*DP84+DP21*P84-0.20512820*DP74   ALMGL397
1812      P95=P21*P85-0.16923076*P75      ALMGL398
1813      DP95=P21*DP85+DP21*P85-0.16923076*DP75   ALMGL399
1814      P96=P21*P86-0.12307692*P76      ALMGL400
1815      DP96=P21*DP86+DP21*P86-0.12307692*DP76   ALMGL401
1816      P97=P21*P87-0.06666666*P77      ALMGL402
1817      DP97=P21*DP87+DP21*P87-0.06666666*DP77   ALMGL403
1818      P98=P21*P88
1819      DP98=P21*DP88+DP21*P88
1820      P99=P22*P88

```

1821	DP99=8.0*P98	ALMGL412
1822	AOR=AOR*AR	ALMGL413
1823	C2=G(9,2)*CP2+G(1,9)*SP2	ALMGL414
1824	C3=G(9,3)*CP3+G(2,9)*SP3	ALMGL415
1825	C4=G(9,4)*CP4+G(3,9)*SP4	ALMGL416
1826	C5=G(9,5)*CP5+G(4,9)*SP5	ALMGL417
1827	C6=G(9,6)*CP6+G(5,9)*SP6	ALMGL418
1828	C7=G(9,7)*CP7+G(6,9)*SP7	ALMGL419
1829	C8=G(9,8)*CP8+G(7,9)*SP8	ALMGL420
1830	C9=G(9,9)*CP9+G(8,9)*SP9	ALMGL421
1831	BR=BR-9.0*AOR*(G(9,1)*P91+C2*P92+C3*P93+C4*P94+C5*P95+C6*P96+C7*P97)	ALMGL422
1832	17+C8*P98+C9*P99)	ALMGL423
1833	BT=BT+AOR*(G(9,1)*DP91+C2*DP92+C3*DP93+C4*DP94+C5*DP95+C6*DP96+C7*DP97)	ALMGL424
1834	1DP97+C8*DP98+C9*DP99)	ALMGL425
1835	BP=BP-AOR*((G(9,2)*SP2-G(1,9)*CP2)*P92+2.0*(G(9,3)*SP3-G(2,9)*CP3))	ALMGL426
1836	1*P93+3.0*(G(9,4)*SP4-G(3,9)*CP4)*P94+4.0*(G(9,5)*SP5-G(4,9)*CP5)*P95	ALMGL427
1837	295+5.0*(G(9,6)*SP6-G(5,9)*CP6)*P96+6.0*(G(9,7)*SP7-G(6,9)*CP7)*P97	ALMGL428
1838	3+7.0*(G(9,8)*SP8-G(7,9)*CP8)*P98+8.0*(G(9,9)*SP9-G(8,9)*CP9)*P99)	ALMGL429
1839	IF (NMAX.LE. 9) GO TO 1	ALMGL430
1840	C	N=10
1841	SP10=SP2*CP9+CP2*SP9	ALMGL431
1842	CP10=CP2*CP9-SP2*SP9	ALMGL432
1843	P101=P21*P91-0.25098039*P81	ALMGL433
1844	DP101=P21*DP91+DP21*P91-0.25098039*DP81	ALMGL434
1845	P102=P21*P92-0.24705882*P82	ALMGL435
1846	DP102=P21*DP92+DP21*P92-0.24705882*DP82	ALMGL436
1847	P103=P21*P93-0.23529411*P83	ALMGL437
1848	DP103=P21*DP93+DP21*P93-0.23529411*DP83	ALMGL438
1849	P104=P21*P94-0.21568627*P84	ALMGL439
1850	DP104=P21*DP94+DP21*P94-0.21568627*DP84	ALMGL440
1851	P105=P21*P95-0.18823529*P85	ALMGL441
1852	DP105=P21*DP95+DP21*P95-0.18823529*DP85	ALMGL442
1853	P106=P21*P96-0.15294117*P86	ALMGL443
1854	DP106=P21*DP96+DP21*P96-0.15294117*DP86	ALMGL444
1855	P107=P21*P97-0.10980392*P87	ALMGL445
1856	DP107=P21*DP97+DP21*P97-0.10980392*DP87	ALMGL446
1857	P108=P21*P98-0.05882352*P88	ALMGL447
1858	DP108=P21*DP98+DP21*P98-0.05882352*DP88	ALMGL448
1859	P109=P21*P99	ALMGL449
1860	DP109=P21*DP99+DP21*P99	ALMGL450
1861	P1010=P22*P99	ALMGL451
1862	DP1010=9.0*P109	ALMGL452
1863	AOR=AOR*AR	ALMGL453
1864	C2=G(10,2)*CP2+G(1,10)*SP2	ALMGL454
1865	C3=G(10,3)*CP3+G(2,10)*SP3	ALMGL455
1866	C4=G(10,4)*CP4+G(3,10)*SP4	ALMGL456
1867	C5=G(10,5)*CP5+G(4,10)*SP5	ALMGL457
1868	C6=G(10,6)*CP6+G(5,10)*SP6	ALMGL458
1869	C7=G(10,7)*CP7+G(6,10)*SP7	ALMGL459
1870	C8=G(10,8)*CP8+G(7,10)*SP8	ALMGL460
1871	C9=G(10,9)*CP9+G(8,10)*SP9	ALMGL461
1872	C10=G(10,10)*CP10+G(9,10)*SP10	ALMGL462
1873	BR=BR-10.0*AOR*(G(10,1)*P101+C2*P102+C3*P103+C4*P104+C5*P105+C6*P106)	ALMGL463
1874	106+C7*P107+C8*P108+C9*P109+C10*P1010)	ALMGL464
1875	BT=BT+AOR*(G(10,1)*DP101+C2*DP102+C3*DP103+C4*DP104+C5*DP105+C6*DP106)	ALMGL465
1876	1106+C7*DP107+C8*DP108+C9*DP109+C10*DP1010)	ALMGL466
1877	BP=BP-AOR*((G(10,2)*SP2-G(1,10)*CP2)*P102+2.0*(G(10,3)*SP3-G(2,10)*CP3))	ALMGL467
1878	1*CP3)*P103+3.0*(G(10,4)*SP4-G(3,10)*CP4)*P104+4.0*(G(10,5)*SP5-G(4,10)*CP5)*P105	ALMGL468
1879	2,10)*CP6)*P106+6.0*(G(10,7)*SP7-G(6,10)*CP7)*P107+7.0*(G(10,8)*SP8-G(7,10)*CP8)*P108+8.0*(G(10,9)*SP9-G(8,10)*CP9)*P109+9.0*(G(10,10)*SP10-G(9,10)*CP10)*P1010)	ALMGL469
1880	4*SP9-G(8,10)*CP9)*P109+9.0*(G(10,10)*SP10-G(9,10)*CP10)*P1010)	ALMGL470
1881	IF (NMAX.LE.10) GO TO 1	ALMGL471
1882	C	N=11
1883	SP11=(SP6+SP6)*CP6	ALMGL472
1884	CP11=(CP6+SP6)*(CP6-SP6)	ALMGL473
1885	P111=P21*P101-0.25077399*P91	ALMGL474
1886	DP111=P21*DP101+DP21*P101-0.25077399*DP91	ALMGL475
1887	P112=P21*P102-0.24767801*P92	ALMGL476
1888	DP112=P21*DP102+DP21*P102-0.24767801*DP92	ALMGL477
1889	P113=P21*P103-0.23839009*P93	ALMGL478
1890		ALMGL479

1891	DP113=P21*DP103+DP21*P103-0.23839009*DP93	ALMGL482
1892	P114=P21*P104-0.22291021*P94	ALMGL483
1893	DP114=P21*DP104+DP21*P104-0.22291021*DP94	ALMGL484
1894	P115=P21*P105-0.20123839*P95	ALMGL485
1895	DP115=P21*DP105+DP21*P105-0.20123839*DP95	ALMGL486
1896	P116=P21*P106-0.17337461*P96	ALMGL487
1897	DP116=P21*DP106+DP21*P106-0.17337461*DP96	ALMGL488
1898	P117=P21*P107-0.13931888*P97	ALMGL489
1899	DP117=P21*DP107+DP21*P107-0.13931888*DP97	ALMGL490
1900	P118=P21*P108-0.09907120*P98	ALMGL491
1901	DP118=P21*DP108+DP21*P108-0.09907120*DP98	ALMGL492
1902	P119=P21*P109-0.05263157*P99	ALMGL493
1903	DP119=P21*DP109+DP21*P109-0.05263157*DP99	ALMGL494
1904	P1110=P21*P1010	ALMGL495
1905	DP1110=P21*DP1010+DP21*P1010	ALMGL496
1906	P1111=P22*P1010	ALMGL497
1907	DP1111=10.0*P1110	ALMGL498
1908	AOR=AOR*AR	ALMGL499
1909	C2=G(11,2)*CP2+G(1,11)*SP2	ALMGL500
1910	C3=G(11,3)*CP3+G(2,11)*SP3	ALMGL501
1911	C4=G(11,4)*CP4+G(3,11)*SP4	ALMGL502
1912	C5=G(11,5)*CP5+G(4,11)*SP5	ALMGL503
1913	C6=G(11,6)*CP6+G(5,11)*SP6	ALMGL504
1914	C7=G(11,7)*CP7+G(6,11)*SP7	ALMGL505
1915	C8=G(11,8)*CP8+G(7,11)*SP8	ALMGL506
1916	C9=G(11,9)*CP9+G(8,11)*SP9	ALMGL507
1917	C10=G(11,10)*CP10+G(9,11)*SP10	ALMGL508
1918	C11=G(11,11)*CP11+G(10,11)*SP11	ALMGL509
1919	BR=BR-11.0*AOR*(G(11,1)*P111+C2*P112+C3*P113+C4*P114+C5*P115+C6*P116+C7*P117+C8*P118+C9*P119+C10*P110+C11*P111)	ALMGL510
1920	1116+C7*P117+C8*P118+C9*P119+C10*P110+C11*P111)	ALMGL511
1921	BT=BT+AOR*(G(11,1)*DP111+C2*DP112+C3*DP113+C4*DP114+C5*DP115+C6*DP116+C7*DP117+C8*DP118+C9*DP119+C10*DP110+C11*DP111)	ALMGL512
1922	1116+C7*DP117+C8*DP118+C9*DP119+C10*DP110+C11*DP111)	ALMGL513
1923	BP=BP-AOR*((G(11,2)*SP2-G(1,11)*CP2)*P112+2.0*(G(11,3)*SP3-G(2,11)*CP3)*P113+3.0*(G(11,4)*SP4-G(3,11)*CP4)*P114+4.0*(G(11,5)*SP5-G(4,11)*CP5)*P115+5.0*(G(11,6)*SP6-G(5,11)*CP6)*P116+6.0*(G(11,7)*SP7)*P117+7.0*(G(11,8)*SP8-G(7,11)*CP8)*P118+8.0*(G(11,9)*SP9-G(8,11)*CP9)*P119+9.0*(G(11,10)*SP10-G(9,11)*CP10)*P110+10.0*(G(11,11)*SP11-G(10,11)*CP11)*P111)	ALMGL514
1924	5*(G(11,11)*SP11-G(10,11)*CP11)*P111)	ALMGL515
1925	IF (NMAX.LE.11) GO TO 1	N=12
1926		ALMGL516
1927		ALMGL517
1928		ALMGL518
1929		ALMGL519
1930		ALMGL520
C		ALMGL521
1931	SP12=SP2*CP11+CP2*SP11	ALMGL522
1932	CP12=CP2*CP11-SP2*SP11	ALMGL523
1933	P121=P21*P111-0.25062656*P101	ALMGL524
1934	DP121=P21*DP111+DP21*P111-0.25062656*DP101	ALMGL525
1935	P122=P21*P112-0.24812030*P102	ALMGL526
1936	DP122=P21*DP112+DP21*P112-0.24812030*DP102	ALMGL527
1937	P123=P21*P113-0.24060150*P103	ALMGL528
1938	DP123=P21*DP113+DP21*P113-0.24060150*DP103	ALMGL529
1939	P124=P21*P114-0.22807017*P104	ALMGL530
1940	DP124=P21*DP114+DP21*P114-0.22807017*DP104	ALMGL531
1941	P125=P21*P115-0.21052631*P105	ALMGL532
1942	DP125=P21*DP115+DP21*P115-0.21052631*DP105	ALMGL533
1943	P126=P21*P116-0.18796992*P106	ALMGL534
1944	DP126=P21*DP116+DP21*P116-0.18796992*DP106	ALMGL535
1945	P127=P21*P117-0.16040100*P107	ALMGL536
1946	DP127=P21*DP117+DP21*P117-0.16040100*DP107	ALMGL537
1947	P128=P21*P118-0.12781954*P108	ALMGL538
1948	DP128=P21*DP118+DP21*P118-0.12781954*DP108	ALMGL539
1949	P129=P21*P119-0.09022556*P109	ALMGL540
1950	DP129=P21*DP119+DP21*P119-0.09022556*DP109	ALMGL541
1951	P1210=P21*P110-0.04761904*P1010	ALMGL542
1952	DP1210=P21*DP110+DP21*P110-0.04761904*DP1010	ALMGL543
1953	P1211=P21*P111	ALMGL544
1954	DP1211=P21*DP111+DP21*P111	ALMGL545
1955	P1212=P22*P111	ALMGL546
1956	DP1212=11.0*P1211	ALMGL547
1957	AOR=AOR*AR	ALMGL548
1958	C2=G(12,2)*CP2+G(1,12)*SP2	ALMGL549
1959	C3=G(12,3)*CP3+G(2,12)*SP3	ALMGL550
1960	C4=G(12,4)*CP4+G(3,12)*SP4	ALMGL551

```

1961      C5=G(12,5)*CP5+G(4,12)*SP5          ALMGL552
1962      C6=G(12,6)*CP6+G(5,12)*SP6          ALMGL553
1963      C7=G(12,7)*CP7+G(6,12)*SP7          ALMGL554
1964      C8=G(12,8)*CP8+G(7,12)*SP8          ALMGL555
1965      C9=G(12,9)*CP9+G(8,12)*SP9          ALMGL556
1966      C10=G(12,10)*CP10+G(9,12)*SP10        ALMGL557
1967      C11=G(12,11)*CP11+G(10,12)*SP11        ALMGL558
1968      C12=G(12,12)*CP12+G(11,12)*SP12        ALMGL559
1969      BR=BR-12.0*AOR*(G(12,1)*P121+C2*P122+C3*P123+C4*P124+C5*P125+C6*P126+          ALMGL560
1970      126+C7*P127+C8*P128+C9*P129+C10*P1210+C11*P1211+C12*P1212)          ALMGL561
1971      BT=BT+AOR*(G(12,1)*DP121+C2*DP122+C3*DP123+C4*DP124+C5*DP125+C6*DP126+          ALMGL562
1972      1126+C7*DP127+C8*DP128+C9*DP129+C10*DP1210+C11*DP1211+C12*DP1212)          ALMGL563
1973      BP=BP-AOR*((G(12,2)*SP2-G(1,12)*CP2)*P122+2.0*(G(12,3)*SP3-G(2,12))          ALMGL564
1974      1*CP3)*P123+3.0*(G(12,4)*SP4-G(3,12)*CP4)*P124+4.0*(G(12,5)*SP5-G(4,12))          ALMGL565
1975      2,12)*CP5)*P125+5.0*(G(12,6)*SP6-G(5,12)*CP6)*P126+6.0*(G(12,7)*SP7)          ALMGL566
1976      3-G(6,12)*CP7)*P127+7.0*(G(12,8)*SP8-G(7,12)*CP8)*P128+8.0*(G(12,9)          ALMGL567
1977      4*SP9-G(8,12)*CP9)*P129+9.0*(G(12,10)*SP10-G(9,12)*CP10)*P1210+10.0          ALMGL568
1978      5*(G(12,11)*SP11-G(10,12)*CP11)*P1211+11.0*(G(12,12)*SP12-G(11,12))          ALMGL569
1979      6CP12)*P1212)          ALMGL570
1980      IF (NMAX.LE.12) GO TO 1          N=13          ALMGL571
1981      C
1982      SP13=(SP7+SP7)*CP7          ALMGL572
1983      CP13=(CP7+SP7)*(CP7-SP7)          ALMGL573
1984      P131=P21*P121-0.25051759*P111          ALMGL574
1985      DP131=P21*DP121+DP21*P121-0.25051759*DP111          ALMGL575
1986      P132=P21*P122-0.24844720*P112          ALMGL576
1987      DP132=P21*DP122+DP21*P122-0.24844720*DP112          ALMGL577
1988      P133=P21*P123-0.24223602*P113          ALMGL578
1989      DP133=P21*DP123+DP21*P123-0.24223602*DP113          ALMGL579
1990      P134=P21*P124-0.23188405*P114          ALMGL580
1991      DP134=P21*DP124+DP21*P124-0.23188405*DP114          ALMGL581
1992      P135=P21*P125-0.21739130*P115          ALMGL582
1993      DP135=P21*DP125+DP21*P125-0.21739130*DP115          ALMGL583
1994      P136=P21*P126-0.19875776*P116          ALMGL584
1995      DP136=P21*DP126+DP21*P126-0.19875776*DP116          ALMGL585
1996      P137=P21*P127-0.17598343*P117          ALMGL586
1997      DP137=P21*DP127+DP21*P127-0.17598343*DP117          ALMGL587
1998      P138=P21*P128-0.14906832*P118          ALMGL588
1999      DP138=P21*DP128+DP21*P128-0.14906832*DP118          ALMGL589
2000      P139=P21*P129-0.11801242*P119          ALMGL590
2001      DP139=P21*DP129+DP21*P129-0.11801242*DP119          ALMGL591
2002      P1310=P21*P1210-0.08281573*P1110          ALMGL592
2003      DP1310=P21*DP1210+DP21*P1210-0.08281573*DP1110          ALMGL593
2004      P1311=P21*P1211-0.04347826*P1111          ALMGL594
2005      DP1311=P21*DP1211+DP21*P1211-0.04347826*DP1111          ALMGL595
2006      P1312=P21*P1212          ALMGL596
2007      DP1312=P21*DP1212+DP21*P1212          ALMGL597
2008      P1313=P22*P1212          ALMGL598
2009      DP1313=12.0*P1312          ALMGL599
2010      AOR=AOR*AR          ALMGL600
2011      C2=G(13,2)*CP2+G(1,13)*SP2          ALMGL601
2012      C3=G(13,3)*CP3+G(2,13)*SP3          ALMGL602
2013      C4=G(13,4)*CP4+G(3,13)*SP4          ALMGL603
2014      C5=G(13,5)*CP5+G(4,13)*SP5          ALMGL604
2015      C6=G(13,6)*CP6+G(5,13)*SP6          ALMGL605
2016      C7=G(13,7)*CP7+G(6,13)*SP7          ALMGL606
2017      C8=G(13,8)*CP8+G(7,13)*SP8          ALMGL607
2018      C9=G(13,9)*CP9+G(8,13)*SP9          ALMGL608
2019      C10=G(13,10)*CP10+G(9,13)*SP10        ALMGL609
2020      C11=G(13,11)*CP11+G(10,13)*SP11        ALMGL610
2021      C12=G(13,12)*CP12+G(11,13)*SP12        ALMGL611
2022      C13=G(13,13)*CP13+G(12,13)*SP13        ALMGL612
2023      BR=BR-13.0*AOR*(G(13,1)*P131+C2*P132+C3*P133+C4*P134+C5*P135+C6*P136+          ALMGL613
2024      136+C7*P137+C8*P138+C9*P139+C10*P1310+C11*P1311+C12*P1312+C13*P1313)          ALMGL614
2025      2)
2026      BT=BT+AOR*(G(13,1)*DP131+C2*DP132+C3*DP133+C4*DP134+C5*DP135+C6*DP136+          ALMGL615
2027      1136+C7*DP137+C8*DP138+C9*DP139+C10*DP1310+C11*DP1311+C12*DP1312+C13*          ALMGL616
2028      23*DP1313)
2029      BP=BP-AOR*((G(13,2)*SP2-G(1,13)*CP2)*P132+2.0*(G(13,3)*SP3-G(2,13))          ALMGL617
2030      1*CP3)*P133+3.0*(G(13,4)*SP4-G(3,13)*CP4)*P134+4.0*(G(13,5)*SP5-G(4,13))          ALMGL618

```

```
2031      2,13)*CP5)*P135+5.0*(G(13,6)*SP6-G(5,13)*CP6)*P136+6.0*(G(13,7)*SP7ALMGL622
2032      3-G(6,13)*CP7)*P137+7.0*(G(13,8)*SP8-G(7,13)*CP8)*P138+8.0*(G(13,9)ALMGL623
2033      4*SP9-G(8,13)*CP9)*P139+9.0*(G(13,10)*SP10-G(9,13)*CP10)*P1310+10.0ALMGL624
2034      5*(G(13,11)*SP11-G(10,13)*CP11)*P1311+11.0*(G(13,12)*SP12-G(11,13)*ALMGL625
2035      6CP12)*P1312+12.0*(G(13,13)*SP13-G(12,13)*CP13)*P1313)ALMGL626
2036      1 BR = BR / 100000. ALMGL627
2037      BT = BT / 100000. ALMGL628
2038      BP = BP / ST / 100000. ALMGL629
2039      B = DSQRT(BR*BR+BT*BT+BP*BP) ALMGL632
2040      RETURN ALMGL634
2041      END ALMGL635
```